INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Diplomarbeit

Konzeption eines IT-Forensikleitfadens für das Leibniz-Rechenzentrum

Anton Romanyuk

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Diplomarbeit

Konzeption eines IT-Forensikleitfadens für das Leibniz-Rechenzentrum

Anton Romanyuk

Aufgabensteller: Dr. Helmut Reiser

Betreuer: Dr. Wolfgang Hommel
Stefan Metzger

Abgabetermin: 13. Januar 2012

Hiermit versichere ich, dass ich die vorliegende Diplomarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 13. Januar 2012

. . . . . .

(Unterschrift des Kandidaten)

In Zeiten zunehmend komplexer IT-Infrastrukturen und wachsender Sicherheitsbedrohungen spielt der Einsatz von modernen Sicherheitslösungen für die IT-Systeme eine immer größere Rolle. Es existieren diverse Szenarien, die zum völligen Erliegen der IT-gestützten Prozesse führen können. Die Aufklärung von erfolgten Angriffen verlangt den Unternehmen immer mehr Ressourcen ab und setzt großes Know-how und Erfahrung voraus. Immer mehr Unternehmen erweitern ihr Sicherheitskonzept um forensische Untersuchungen.

Das ist eine Herausforderung, der sich auch das Leibniz-Rechenzentrum (LRZ) mit Sitz in Garching stellen muss. Im Rahmen dieser Diplomarbeit wurde ein praxistauglicher IT-Forensik Leitfaden für Linux- und Windows-Server-Systeme am LRZ erstellt.

Es wurde beleuchtet, wo man bei einem Security-Incident nach Beweisen suchen sollte, wie man sie erkennen kann, wie sie zu bewerten sind und wie sie für juristische Täterverfolgung verwertbar gesichert werden sollten - alles unter der Berücksichtigung der komplexen IT-Infrastruktur des Leibniz-Rechenzentrums und zahlreich eingesetzter Sicherheitsmaßnahmen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1. Einleitung

Insider wissen es längst: IT-Forensik wird in den kommenden Jahren eines der Top-Themen der IT-Industrie. Sie sorgt für ganz neue Einblicke und Sichtweisen bei der Bewältigung von typischen Sicherheitsvorfällen. Schon heute profitieren viele Firmen von der IT-Forensik. Das gibt konkret Anlass dazu, sich gezielt mit der Thematik auseinanderzusetzen.

1.1. Computer-Forensik: Methodisches Vorgehen als Schlüssel

Die Bedrohung durch Cyber-Kriminalität hat in den letzten Jahren ihr Gesicht verändert. Mittlerweile sind international organisierte Banden aktiv, die erstaunliches Know-how und beeindruckende Kreativität einsetzen, um Unternehmen zu schädigen. Der Schaden dieses Treibens ist beträchtlich. Laut der von Steria Mummert Consulting in Zusammenarbeit mit dem F.A.Z.-Institut erstellten Studie „Branchenkompass 2011 Telekommunikation´´, beträgt der Schaden für die deutsche Wirtschaft mehr als zehn Milliarden Euro pro Jahr, jedes vierte Unternehmen war in den vergangenen Jahren Opfer eines Cyberangriffs [Mum]. Anlässlich der Eröffnung des IT-Sicherheitskongresses 2011 berichtete Michael Hange, der Präsident des Bundesamtes für Sicherheit in der Informationstechnik, dass in Deutschland täglich 20 000 Webseiten neu infiziert werden. Er beklagte, dass die Wirtschaftsspionage aus dem Ausland bereits Formen eines Wirtschaftskrieges angenommen hat und dass in vielen Unternehmen und Behörden das Sicherheitsbewusstsein leider noch zu wünschen übrig lässt. Reagiert wird leider häufig erst nach einem großen Angriff. [VB]

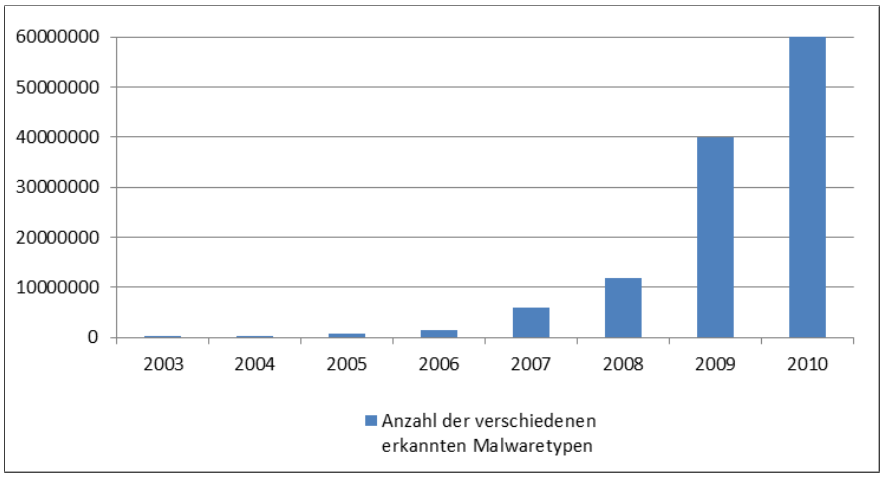

Abbildung 1.1.: Entwicklung der Malware 2003-2010.

Quelle: [Pan10, S. 19]

Der aktuelle Jahresreport (2010) von Panda Labs [Pan10] zeichnet ebenfalls ein düsteres Bild der aktuellen Sicherheitslage. Aus den Unterlagen geht hervor, dass im Zeitraum zwischen Januar und Dezember 2010 34% aller jemals entwickelten Schädlinge erschienen. Insgesamt wurden 134 Millionen Dateien analysiert, von denen 60 Millionen als Malware klassifiziert wurden. Noch schwerer wiegt aber, dass knapp 20 Millionen der untersuchten Dateien unbekannte Malware-Muster enthielten. Damit sieht Panda Labs die Tendenz der letzten Jahre als fortgesetzt und spricht ebenfalls von einem globalen Wirtschaftskrieg, der über illegale Machenschaften im Netz ausgetragen wird.

Das bekannteste Beispiel der jüngsten Vergangenheit dürfte „Stuxnet´´ sein, ein Windows-Wurm, der sich über USB-Sticks verbreitet und durch den erfolgreichen Angriff auf das iranische Atomkraftwerk Buschehr weltweite Bekanntheit erlangte. Beinahe gleichzeitig verbreitete sich der „Here you have"-Wurm wie ein Lauffeuer über viele Systeme auf eine Art, die stark an die „Anna Kournikova"-Virusepidemie, die circa zehn Jahre zurück liegt, erinnerte. Der E-Mail-Virus wurde von der islamistischen Terroristenvereinigung „Brigades of Tariq ibn Ziyad" entwickelt als Antwort auf die geplante Verbrennung des Korans durch den US-Pastor Terry Jones. Als ein weiterer Negativhöhepunkt des Jahres 2010 gelten von McAfee als „Operation Aurora" bezeichneten Cyberattacken auf US-Unternehmen, eine breit angelegte Wirtschaftsspionage aus China.

Diese Angriffe markieren einen Wendepunkt im Bereich Cyber-Security: Die ehemalige Hacker-Szene hat sich in den letzten Jahren zu einer sehr professionell und innovativ arbeitenden Industrie entwickelt, die zudem auch ziemlich profitabel ist.

In Zeiten zunehmend komplexer IT-Infrastrukturen und wachsender Sicherheitsbedrohungen spielt der Einsatz von modernen Sicherheitslösungen für die IT-Systeme eine immer größere Rolle. Es existieren diverse Szenarien, die zum völligen Erliegen der IT-gestützten Prozesse führen können. Die Abwehr von aktuellen Bedrohungen verlangt den Unternehmen immer mehr Ressourcen ab und setzt großes Know-how und Erfahrung voraus. Die Cyber-Angriffe haben mittlerweile eine so hohe Komplexität erreicht, dass meist kombinierte Maßnahmen für die Aufrechterhaltung der Sicherheit notwendig sind. Firewalls und Antivirenprogramme allein reichen aktuell nicht mehr aus und werden mit Intrusion-Detection-Systemen (IDS) ergänzt. Unternehmen gehen ferner dazu über, alle Daten automatisch zu analysieren, um Sicherheitslöcher bereits im Vorfeld zu erkennen und schließen zu können. Außerdem ist es inzwischen zwingend für immer mehr Unternehmen, ihr Sicherheitskonzept um forensische Untersuchungen (Computer-Forensik bzw. IT-Forensik) zu erweitern.

> *IT-Forensik ist die streng methodisch vorgenommene Untersuchung auf Datenträgern und in Computernetzen zum Nachweis und zur Aufklärung von Straftaten im Zusammenhang mit IT-Systemen. IT-Forensik stellt dabei den Kontext zwischen der Incident Response und einer erfolgreichen Strafverfolgung dar.*

IT-Forensik ist seit dem letzten Jahrzehnt ganz oben auf der Agenda bei IT-Themen, und in dem Maße, wie die technischen Möglichkeiten der Hacker zunahmen, wuchs auch die Bedeutung für die Verbesserung der Abwehrmaßnahmen.

Ziel der Computer-Forensik ist es, das Eintreten von diversen Szenarien vorauszusehen, die zum Erliegen von IT führen könnten, und sich methodisch darauf vorzubereiten. IT-Prozesse und andere auf IT-basierende Prozesse können im Worst-Case nicht fortgeführt werden, daher sollen Sicherheitsvorfälle zuverlässig erkannt und ihre Auswirkungen eingedämmt werden. Die strategische Vorbereitung unter Einbeziehung der Computer-Forensik bildet die Grundlage für eine systematische Lösungskonzeption, um beim Eintreten eines Incidents die Handlungsfähigkeit zu erhalten und gleichzeitig den Zeitaufwand für die Fallbearbeitung zu reduzieren. Ferner sollen bei der Incident-Bearbeitung genügend Beweise zur Täterermittlung und weiteren juristischen Verfolgung gesammelt werden.

Längst fokussiert sich Computer-Forensik nicht mehr nur auf den reinen Ermittlungsprozess, sondern deckt den IT-Alltag sowie die kontinuierliche Verbesserung des Incident-Response-Prozesses ab. Eine klar definierte, nach Geschäftsanforderungen abgeglichene IT-Strategie bildet die Voraussetzung für eine zukunftsfähige IT-Organisation.

Das ist eine Herausforderung, der sich auch das Leibniz-Rechenzentrum (LRZ) mit Sitz in Garching stellen muss. Mit seinen unterschiedlichen IT-Dienstleistungen zählt das LRZ zu den größten wissenschaftlichen Rechenzentren in Deutschland und betreibt Hochleistungsrechensysteme für alle bayerischen Hochschulen, sowie einen Bundeshöchstleistungsrechner, der allen deutschen Hochschulen zur Verfügung steht. Insgesamt nehmen über 120.000 Anwender den Service des Leibniz-Rechenzentrums in Anspruch und werden dabei von „nur" rund 150 von Mitarbeitern betreut.

1.2. Aufgabenstellung: IT-Forensik-Leitfaden maßgeschneidert

Computer-Forensik umfasst wesentlich mehr Aspekte, als man auf den ersten Blick glauben könnte. Es existieren bereits Untergliederungen der Disziplin, wie zum Beispiel die „mobile Forensik“ oder die „Netzwerk-Forensik“. Nicht umsonst ist der „IT-Forensik-Leitfaden“ des Bundesamtes für Sicherheit in der Informationstechnik (BSI) in der derzeit gültigen Version 1.0.1 auf 354 Seiten angewachsen. Je komplexer und zahlreicher die Sicherheitsmaßnahmen, desto höher ist der zeitliche Aufwand und die damit verbundenen Kosten. Betriebssysteme und die Hardware müssen gewartet werden, zudem erfordert die dauerhafte Weiterentwicklung der Ermittlungstechniken und das stetig vielfältiger werdende Angebot an Forensik-Tools immer neue Investitionen in die Verfeinerung der internen Sicherheitsprozesse, eingesetzten Technologien und die Weiterbildung des Personals. Die in den aktuellen IT-Publikationen erhältliche Sammlungen von Ermittlungstechniken sind zu umfangreich, um die Service-Bereitstellung am LRZ exakt abbilden zu können.

Die Aufgabe dieser Diplomarbeit ist die Erstellung eines praxistauglichen IT-Forensik Leitfadens für Linux- und Windows-Server-Systeme am LRZ. Es soll beleuchtet werden, wo man bei einem Security-Incident nach Beweisen suchen sollte, wie man sie erkennen kann, wie sie zu bewerten sind und wie sie für juristische Täterverfolgung verwertbar gesichert werden sollten. Des Weiteren sollen für den Ermittlungsprozess relevante juristische Aspekte beleuchtet werden. Die weiterführenden Bereiche der Computer-Forensik werden dagegen nur dann erwähnt, wenn sie für das technische Verständnis Relevanz besitzen.

Die Erprobung des Leitfadens ist, im Anschluss an eine Konzeption und die Vorbereitung von Ermittlungsmaßnahmen, unverzichtbar. Ein IT-Forensik-Leitfaden ist im Ernstfall wirkungslos, wenn die Wirksamkeit der Maßnahmen nicht verifiziert wird.

Im Anschluss an die Diplomarbeit sollte der Leitfaden im Rahmen eines kontinuierlichen Verbesserungsprozesses verankert werden, um fortlaufend Optimierungspotenziale aufzudecken und umsetzen zu können. So sollten neue oder alternative Vorgehensweise in den Ermittlungsprozess eingebaut werden.

1.3. Gliederung der Arbeit

Die Arbeit gliedert sich im Wesentlichen in acht Kapitel, die sich jeweils einem Themengebiet widmen. Jeder Abschnitt ist in sich abgeschlossen und kann separat gelesen werden.

Kapitel 2: Analyse der aktuellen Bedrohungssituation und typischer Angriffsszenarien

In diesem Kapitel werden der Wandel der Bedrohungssituation im Internet und die Protagonisten in der Malware-Szene beleuchtet. Weiterhin findet sich dort eine Analyse der geläufigen Angriffstechniken.

Kapitel 3: Security-Incident-Response-Prozess am LRZ

Kapitel 3 erklärt die wesentlichen Bestandteile des am LRZ eingesetzten Ansatzes für das Management von Sicherheitsvorfällen. Es werden diverse zum Einsatz kommende präventive, detektierende und reaktive Maßnahmen vorgestellt und der typische Ablauf des Incident-Response-Prozesses am LRZ erklärt. Die Erläuterungen gehen aber nur so weit, wie sie für das Verständnis der Zusammenhänge benötigt werden.

Kapitel 4: Einführung in die Computer-Forensik

Die wesentlichen Grundlagen der IT-Forensik sind Kapitel 4 zu entnehmen. Ziele der Computer-Forensik, organisatorische wie auch technische Anforderungen an den Ermittlungsprozess, als auch die typischen Phasen der Ermittlung stehen im Mittelpunkt dieses Kapitels.

Kapitel 5: Forensische Datenerfassung

Kapitel 5 erklärt im Überblick den Prozess der forensischen Beweismittelsicherung. Hierzu gehören unter anderem das Sammeln von Daten vom noch lebenden System, Datenerfassung im Netzwerk sowie das Anfertigen von forensischen Kopien von Datenträgern. Es werden bewährte IT-Forensik-Tools vorgestellt sowie an konkreten Beispielen sowohl für SuSe Linux 11 Enterprise als auch für Windows Server 2008 Systeme illustriert. Auch die unterschiedlichen Vorgehensweisen bei virtuellen und physischen Server-Systemen sind

diesem Kapitel zu entnehmen. Die häufigsten Fehler bei der Beweissicherung werden gezeigt und es wird erklärt, wie man diese vermeidet.

Kapitel 6: Forensische Analyse im Fokus

Kapitel 6 beschäftigt sich hauptsächlich mit den technischen Aspekten der forensischen Datenanalyse anhand von praktischen Beispielen. Das umfasst nicht nur das Entdecken sondern auch die richtige Interpretation von Beweisen im Kontext der Ermittlung.

Kapitel 7: Forensischer Bericht

Die notwendige Dokumentation des gesamten Ermittlungsprozesses wird im Kapitel 7 illustriert.

Kapitel 8: Zusammenfassung

Kapitel 8 fasst die wichtigsten Erkenntnisse der Arbeit zusammen. Hierzu gehören nicht nur die wesentlichen Eckpunkte aus den vorangegangenen Kapiteln, sodern auch auch ein Ausblick auf die Optimierungspotenziale des Leitfadens.

Anhang

Anhang liefert speziell auf den Einsatz am LRZ angepasste Vorlagen für Beweismittelsicherung sowie Dokumentation verschiedener Aktionen, die im Laufe des Ermittlungsprozesses anfallen. Außerdem wird eine „Forensik-Check-Liste“ vorgestellt, die den Mitarbeitern des Leibniz-Rechenzentrums ermöglichen soll, sich schnell einen Überblick über den aktuellen Systemstatus und die Aktivitäten des Angreifers zu verschaffen und die passenden Gegenmaßnahmen und Werkzeuge auszuwählen.

2. Analyse der aktuellen Bedrohungssituation und typischer Angriffsszenarien

Es liegt auf der Hand, dass Informationen über aktuelle Bedrohungen und mögliche Angriffsszenarien eine der Grundlagen für eine erfolgreiche Aufklärung von Cyber-Angriffen liefern. Im Rahmen dieses Kapitels werden die wichtigsten Aspekte, die für die IT-Forensik von Interesse sind, knapp beleuchtet. Dabei geht es vorrangig um die Beantwortung folgender Fragen:

- Welche Teile der LRZ-Infrastruktur sind besonders bedroht?
- Wie geht ein Angreifer im Allgemeinen vor?
- Welche Arten von Spuren hinterlässt er dabei?
- Welche Unterschiede gibt es zwischen einem Innen- und einem Außentäter?

2.1. Typische Angriffe am LRZ

Die Identifizierung der möglichen Angriffsszenarien ist notwendig, um technische Abwehrmaßnahmen zu verfeinern, um Angriffe rechtzeitig zu erkennen sowie um bei einem eingetretenen Sicherheitsvorfall richtig reagieren zu können. Das gilt in ganz besonderem Maße für die typischen Angriffsarten, denen das LRZ oft ausgesetzt ist.

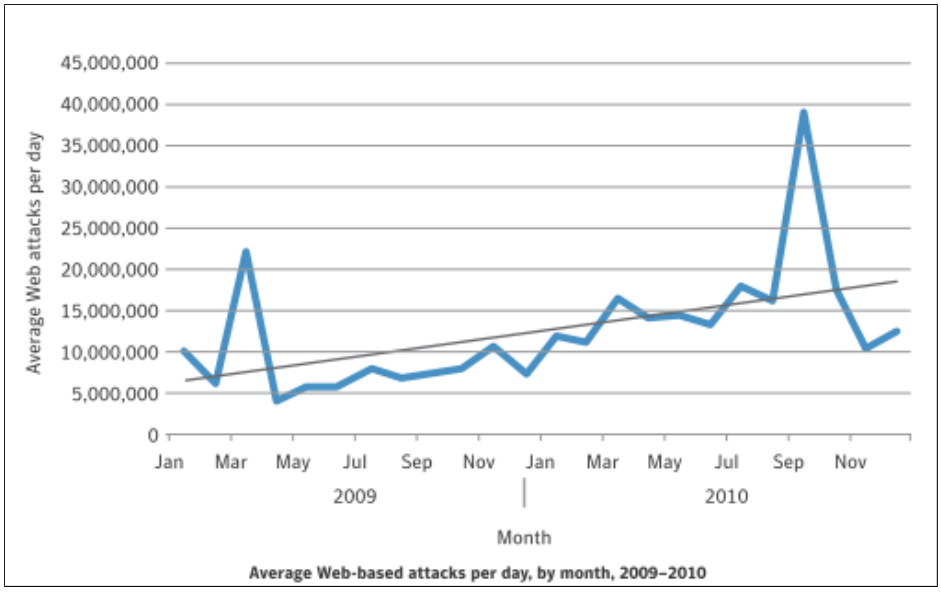

Abbildung 2.1.: Entwicklung der webbasierten Angriffe 2009/2010.

Quelle: [Fos10, S. 13]

Eine Herausforderung ist, dass die Malware-Autoren und Hacker mit enormer krimineller Energie agieren. Der Jahresbericht 2010 der Symantec Corporation [Fos10] zeigt, dass die gezielten Angriffe im Vergleich zum Vorjahr zugenommen haben. Für das Jahr 2011 gehen die Prognosen davon aus, dass die Frequenz der Angriffe weiterhin steigen wird.

Über das gesamte Jahr 2010 betrachtet wurden über 6000 neue Schwachstellen entdeckt: insgesamt war eine Zunahme um 160% im Vergleich zum Vorjahr zu verzeichnen.[vgl. [Fos10]]

Ein weiterer Aspekt der Sicherheitsproblematik ist die rapide Entwicklung der sog. Angriffstoolkits. Der Anteil der mit ihrer Hilfe durchgeführten Angriffe ist 2010 stetig gestiegen und erreichte seinen Spitzenwert im Oktober 2010, als Schätzungen zufolge rund 40 Mio. Angriffe am Tag mit diesen Schadprogrammen durchgeführt wurden (vgl. Abbildung 2.1). Über das gesamte Jahr 2010 betrachtet fand ein Anstieg um knapp 90% statt.

Es ist eine unabdingbare Prämisse für die Reduktion von Sicherheitsrisiken, die typischen Sicherheitsvorfälle am Leibniz-Rechenzentrum einer exakten Analyse zu unterziehen, Schwachstellen der Infrastruktur zu identifizieren und mit geeigneten Maßnahmen darauf zu reagieren.

Typische Angriffsarten umfassen:

SSH Brute-Force SSH Brute-Force-Angriff ist eine altbewährte Methode, um widerrechtlich Remote-Zugriff auf einen Server zu erhalten. Tagtäglich werden interne LRZ-Systeme mit vielen Anmeldeversuchen mit weit verbreiteten Benutzernamen („root“, „admin“, „guest“ o. ä.) in Verbindung mit Wörterbucheinträgen als Passwörter überzogen. Es kommt vor, dass gelegentlich Administratoren beim Vergeben neuer Kennungen simple Passwörter verwenden und somit einem Angreifer den Einstieg in ein System erleichtern. Bei Testsystemen führen, oft bedingt durch eine unsachgemäße Konfiguration der UNIX-Systeme (Wegfall der Zugriffskontrolle in */etc/hosts.allow*) und bereits einfache Username/Passwort-Kombinationen wie „test“/„test“ zum Erfolg.

Root-Exploit Bei den Höchstleistungssystemen am LRZ kommen speziell für das HPC- und Grid-Computing angepasste Kernel zum Einsatz, die auch unabhängig von dem Standard-Kernel gepflegt werden. Nachteilig wirkt sich aus, dass bekannte Sicherheitslücken (auch die, die bereits aktiv ausgenutzt werden), bedingt durch die komplexen Änderungen des Quellcodes, u.U. erst nach einigen Wochen geschlossen werden können. Aus demselben Grund ist auch die Installation von Standard-Patches erschwert. Meist müssen sie vor dem Deployment ausreichend getestet und ggf. modifiziert werden - die betroffenen Rechner sind manchmal Tage bzw. Wochen kompromitierbar. Besonders die Betreiber von Botnetzen sind gezielt auf der Suche nach solchen angreifbaren Systemen. Nach dem erfolgreichen Einstieg in das System kann der Angreifer durch die Anwendung einen Root-Exploits die Zugriffsrestriktionen umgehen, an höhere Rechte gelangen und anschließend schadhaften Code über das Netzwerk nachladen und installieren. Insbesondere Systeme, auf denen mehrere Benutzer parallel arbeiten dürfen, sind in hohem Maße gefährdet, da ein Missbrauch nicht immer sofort auffällt. Ein weiterer, bekannter Root-Exploit, mit dem Angreifer an höhere Rechte gelangen können, beruht auf Diebstahl von SSH-Keys von anderen Rechenzentren. Systeme, auf denen passwortlose Authentifizierung mittels SSH-Key verwendet werden, laufen Gefahr, dass ein Angreifer die Lücke nutzt, um ein System unter seine Kontrolle zu bekommen.

Angriffe auf VoIP-Telefon Nach anfänglicher Skepsis und Anlaufschwierigkeiten setzt sich VoIP nun langsam durch [BW]. Die Juli-August/2011 Ausgabe des E-3 Magazins bescheinigt der VoIP-Technologie gar „Marktreife“ [vgl. [Fär11, S.94]]. Klassische Telefonanlagen werden zunehmend durch IP-basierte Telefonie ersetzt. Die Vorteile von VoIP liegen in erster Linie in der standortunabhängigen Kommunikation. Das macht die VoIP-Technologie zu einem beliebten Ziel der Cyber-Angriffe. In den letzten Jahren nahmen die Angriffe auf VoIP zu, und es wurden immer neue Exploits entdeckt: Sie reichen von Abhören und Umleiten von Anrufen bis hin zu Denial-of-Service-Angriffen und Gebührenbetrug. Das Problem liegt oftmals in den Web-Schnittstellen der VoIP-Software, die zur Konfiguration und Verwaltung des Telefons verwendet werden - ein IP-Telefon ist im Endeffekt nichts anders als ein Computer. Dies kann von Angreifern zum Beispiel für eine Cross-Site-Scripting-Attacke (XSS) oder Cross-Site-Request-Forgeries (XSRF) missbraucht werden, um vollen Zugriff auf das System zu erlangen [Wat09].

Apache Tomcat-Server Durch fehlerhafte Implementierungen von diversen Funktionen in der Tomcat-Serverkomponente ist es einem entfernten Angreifer möglich, Sicherheitsrichtlinien zu umgehen, um beispielsweise eigenen Code über das Netzwerk einzuschleusen und auszuführen. Auch kritisch ist geschicktes Ausnutzen von mehreren Schwachstellen im Apache Tomcat mit dem Ziel der Erlangung der (Tomcat-)Superuser Rechte für das Ausspionieren oder Modifikation von sensiblen Daten.

Virtuelle Apache-Webserver Häufig haben Webserver Schwächen in der Implementierung von PHP im Apache Server. Eine häufig anzutreffende Angriffsart ist das sog. Defacing oder Defacement. Bei dieser speziellen Art des Hacks geht es darum, eine Website zu verunstalten (vgl. Abbildung 2.2). Oft sind Defacements nur als Hinweis auf Schwachstellen aufzufassen.

Abbildung 2.2.: Defacement am Beispiel der Webseite lrz.de.

Analog dazu werden gelegentlich die Index-Seiten modifiziert, um andere Inhalte anzuzeigen, aber auch dazu verwendet, Schadsoftware zu verteilen.

PHPMyAdmin Für die Verwaltung der Datenbanken wird am Leibniz-Rechenzentrum oftmals das grafische Frontend „phpMyAdmin“ eingesetzt. Dabei handelt es sich um eine weitverbreitete serverseitige Anwendung, die auf unixoiden Servern oft vorinstalliert ist. Dementsprechend sind von der Schwachstelle auch unzählige Datenbankverwaltungssysteme betroffen. Fast im Wochentakt werden neue Schwachstellen im „phpMyAdmin“ entdeckt, die zum Teil sehr gefährlich sind, da sie unerlaubten Zugriff auf Verzeichnisse des Servers, Cross-Site-Scripting-Attacken und das Einschleusen von eigenem Code ermöglichen.

2.2. Typischer Angriffsverlauf

Das Wissen über den typischen Angriffsverlauf mag zwar kein Wundermittel sein, ist aber ein sehr wirkungsvolles Instrument, wenn es darum geht, nachzuvollziehen, welche Tätigkeiten der Angreifer auf dem betrof-

fenen System durchgeführt hat und ob mit weiteren Aktionen zu rechnen ist. Mithilfe dieser Informationen können meist Spuren schneller sichergestellt und so zusätzliche Zeitersparnis geschaffen werden. Analog dazu kann auch das Risiko von Folgeangriffen besser eingeschätzt werden. Dadurch können Kosten gesenkt und die Sicherheit der eigenen Netze erhöht werden.

Cyber-Angriffe lassen sich grundsätzlich in mehrere Phasen nach einem Top-Down-Ansatz einteilen. Die Ausprägung der einzelnen Phasen variiert je nach Angriffsart mehr oder weniger stark.

2.2.1. Footprinting

„Footprinting“ umschreibt das Sammeln von Informationen über eine spezifische IT-Infrastruktur meist zum Zweck eines Einbruchsversuchs. Die Informationsbeschaffung selbst lässt sich in zwei Phasen einteilen: passives Footprinting sowie aktives Footprinting.

Passives Footprinting Das Sammeln der Basisinformationen über ein Netzwerk ohne Kontakt mit dem Zielnetzwerk aufzunehmen bezeichnet man als passives Ausspionieren. Der Angreifer versucht Informationen über das Zielobjekt aus öffentlich zugänglichen Quellen zu gewinnen. Unter anderem zählen folgende Schritte zum passiven Ausspionieren:

- Auf der Webseite des Unternehmens bereitgestellten Informationen
- Stellenangebote
- Google Hacking
- Social Engineering

Um an allgemeine Informationen über ein Zielobjekt zu gelangen, bietet sich zunächst ein Blick auf die Unternehmens-Website an. Hier kann ein Angreifer Einzelheiten über die Firma erfahren. Darüber hinaus hat der Hacker die Gelegenheit, Kontaktdaten der Mitarbeiter zu sichern.

Die zweite wichtige Komponente der Vorbereitung stellen die Stellenangebote dar: Die detaillierten Bewerbungsanforderungen sorgen gleichzeitig dafür, dass der Angreifer wertvolle Hinweise über die eingesetzten Betriebssystene, Hardware und Programme gewinnen kann.

Unter „Google Hacking“ (oder auch „Google Dorks“) versteht man elaborierte Google-Abfragen zur Ermittlung der Adressen, die zu vertraulichen Dokumenten, Konfigurationsseiten, Passwortlisten, Backups, Besucherstatistiken und dergleichen führen. Viele Unternehmen sind sich nicht darüber im Klaren, dass über Suchmaschinen (sicherheitsrelevante) Daten entdeckt werden können, die auf der öffentlichen Website nicht verfügbar sind. Hierbei spielt die Eigenschaft der Suchmaschinen eine zentrale Rolle, gefundene Daten unabhängig davon, ob sie versehentlich oder absichtlich zugänglich sind, zu indizieren. Tabelle 2.1 stellt eine kleine Übersicht einiger ausgewählter Google Befehle vor.

Operator	Bedeutung
inurl	Dieser Operator sucht nach Adressen, die den Ausdruck enthalten.
allinurl	Diese Abfragemöglichkeit beschränkt die Ergebnisse auf Adressen, die sämtliche angegebene Suchbegriffe enthalten.
intitle	Diese Abfrage sucht nach Seiten, in deren Titel der Ausdruck vorkommt.
sallintitle	Dieser Operator sucht nach Seiten, die im Titel sämtliche Suchworte enthalten.
filetype	Diese Abfragemöglichkeit beschränkt die Ergebnisse auf Dateien, die die spezifizierte Dateierweiterung haben.

Tabelle 2.1.: Eine Übersicht der erweiterte Suchoperatoren.

Die soziale Komponente spielt eine zunehmend wichtiger werdende Rolle für gezielte Attacken. Erfahrungen aus dem IT-Umfeld zeigen: Ein großer Schwachpunkt in jedem Sicherheitsmodell ist häufig der Mensch. An-

greifer nutzen dabei natürliche, menschliche Reaktionen aus, um an Informationen über eine Zielorganisation zu gelangen. Die Kunst des Täuschens besteht in der vielfältigen Vorgehensweise: Sie reicht von Passwortanfragen bis hin zu Dumpster Diving[1].

Aktives Footprinting Als aktives Ausspionieren wird jene Aktivität bezeichnet, bei der direkt mit dem Zielsystem interagiert wird, um weitere Informationen darüber zu sammeln. Diese Phase dient dem Angreifer zur Identifizierung der interessanten und schlecht abgesicherten Systeme. Informationsquellen sind dabei unter anderem:

- DNS-Abfragen
- Ping Sweep
- Traceroute

DNS-Abfragen - insbesondere des IP-Adress-Bereichs, aus dem eine IP-Adresse stammt - sind notwendig, um eine Liste der IP-Adressen der Zielnetze zu erhalten. Die Grenze zwischen aktiv und passiv ist an dieser Stelle nicht ganz eindeutig: beispielweise lassen sich durch eine WHOIS-Abfrage persönliche Informationen über ein Ziel ermitteln, welche wichtige Ansatzpunkte für beispielsweise Social Engineering bieten könnten, da dort häufig bereits Name, Anschrift, Rufnummern, etc. hinterlegt sind.

Zur Verifikation der erhaltenen IP-Adressen setzen die Angreifer häufig das Ping-Dienstprogramm ein - so können sie schnell und unkompliziert Datenpakete an alle Zielrechner in einem IP-Adress-Bereich senden. Ist ein System aktiv, so antwortet es mit einem Reply-Paket - sofern der Empfang von ICMP-Paketen nicht durch eine Firewall eingeschränkt ist. Die Liste von aktiven Systemen kann anschließend weiter untersucht werden.

```
C:\>tracert wcsaga.com

Routenverfolgung zu wcsaga.com [83.169.6.9] über maximal 30 Abschnitte:

  1     1 ms     1 ms     1 ms  192.168.0.1
  2    38 ms    39 ms    39 ms  lo1.br11.muc.de.hansenet.net [213.191.64.18]
  3    37 ms    38 ms    38 ms  ae0-101.cr02.muc.de.hansenet.net [213.191.88.94]
  4    47 ms    46 ms    46 ms  so-0-3-0-0.cr01.fra.de.hansenet.net [213.191.66.65]
  5    46 ms    46 ms    47 ms  ae1-0.xd01.fra.de.hansenet.net [62.109.69.6]
  6    46 ms    46 ms    46 ms  ae0-0.pr03.decix.de.hansenet.net [213.191.66.138]
  7    75 ms    46 ms    46 ms  ae0.cr-polaris.fra1.he-core.de [80.81.192.239]
  8    49 ms    49 ms    48 ms  xe-0-2-0.cr-nashira.cgn4.he-core.de [80.237.129.109]
  9     *        *        *     Zeitüberschreitung der Anforderung.
 10    50 ms    50 ms    50 ms  www.wingcenter.net [83.169.6.9]

Ablaufverfolgung beendet.
```

Abbildung 2.3.: Traceroute unter Windows

Eine Schlüsselposition bei der Beschaffung von Information über das Zielnetzwerk nimmt das Dienstprogramm Traceroute ein[Abb. 2.3], welches den Weg von Datenpaketen zwischen Quelle und Ziel verfolgt und sichtbar macht. Es ist von der Funktionsweise dem Tool Ping sehr ähnlich, erlaubt es aber, detailliertere Informationen über die Netztopologie zu gewinnen, so dass Angreifer weitere Angaben über geografische Daten und den Service-Provider des Ziels gewinnen können.

2.2.2. Port- und Protokollscan

Der nächste Schritt ist der Port- und Protokollscan. Der Angreifer versucht zu ermitteln, welche der aktiven Systeme offene Ports anbieten und welche Protokolle unterstützt werden, ohne dabei die Monitoring-Mechanismen zu alarmieren. Einige bekannte Scantypen sind unten stehend aufgelistet und im Folgenden kurz erklärt.

- TCP Connect Scan

[1] Beim so genannten Dumpster Diving durchsucht ein Social Engineer die Abfallbehälter systematisch nach Informationen und Anhaltspunkten für das weitere Vorgehen.

- TCP SYN Scan
- TCP FIN Scan/TCP Xmas Scan/TCP Null Scan
- UDP Scan
- ACK Scan

TCP Connect Scan Bei dieser Methode handelt es sich um die einfachste Art des Portscannens. Der Portscanner nutzt das darunterliegende Betriebssystem, indem er eine Systemfunktion connect() aufruft, um sich mit dem Zielhost zu verbinden. Falls der Zielrechner erreichbar ist, wird eine TCP-Verbindung vollständig aufgebaut. Dadurch, dass der sog. Drei-Wege-Handshake zum Einsatz kommt, lässt sich eine solche Verbindung auf dem Zielsystem leicht erkennen und protokollieren.

TCP SYN Scan Im Gegensatz zum „TCP Connect Scan“ wird bei dieser Scan-Methode keine vollständige TCP-Session aufgebaut. Der Scanner beginnt mit einem SYN-Datagramm, so wie es auch im Rahmen des Drei-Wege-Handshake vorgesehen ist. Wird ein RST-Paket empfangen, so lässt das darauf schließen, dass der betroffene Port geschlossen ist. Antwortet der Zielhost allerdings mit einem SYN/ACK- Paket, weiß der Scanner, dass dieser Port offen ist und sendet sofort ein RST-Datagramm, um die im Aufbau befindliche Verbindung wieder zu schließen. Diese Scan-Methode wird immer beliebter, denn viele Systeme protokollieren solche halb-offene Verbindungen nicht. Ferner wird die zur Verfügung stehende Bandbreite sehr effizient ausgenutzt - es lassen sich Tausende Ports pro Sekunde scannen. SYN-Scan wird oft für Denial of Service-Angriffe (SYN-Flooding) benutzt - Angreifer nutzen dabei oft gefälschte IP-Adressen (IP-Spoofing), um Pakete an einer Firewall vorbeizuschleusen. Diese Methode wird auch als „halb-offener“ Scan bezeichnet.

TCP FIN Scan/TCP Xmas Scan/TCP Null Scan Diese Methoden bauen auf die Schwächen der TCP-Protokoll-Spezifikation auf - es wird ausgenutzt, dass geschlossene Ports auf TCP-Datagramme mit einem RST-Paket antworten, während offene Ports TCP-Datagramme verwerfen [RFCc]. Die Unterschiede zwischen den drei Scan-Arten liegen im Detail:

- *TCP FIN Scan* aktiviert das „FIN“-Flag,
- *TCP Xmas Scan* setzt TCP-Flags „FIN“-, „URG“- und „PSH“,
- *TCP Null Scan* aktiviert keine Flags.

Diese auch als „Stealth Scan“ bekannte Methoden sind sehr verbreitet, denn sie werden von Routern und paketfilternden Firewalls u.U. nicht erkannt.

UDP Scan Mit diesem Scan kann man feststellen, welche UDP-Ports offen sind. Ein Scanner sendet hierbei mehrere leere UDP-Datagramme an das Zielsystem. Wird eine „ICMP Port Unreachable“-Fehlermeldung vom Host empfangen, dann kann davon ausgegangen werden, dass der entsprechende Port inaktiv und somit geschlossen ist. Anderenfalls handelt es sich um einen offenen Port. Der verbindungslose Aufbau des UDP-Protokolls birgt ein hohes Potential für falsch positive Resultate. Des Weiteren ist das UDP-Scanning sehr langsam wegen der Limitierung der ICMP-Fehlermeldungsrate durch den RFC 1812 Standard [RFCb].

ACK Scan Zur Identifizierung der Firewall-Funktionsweise wird so genannte ACK Scan verwendet. Dies hat für den Angreifer den großen Vorteil, dass er in der Lage ist, die Firewall- Filtertechnologien zu bestimmen - heutzutage kommen meistens entweder ein einfacher „Packetfilter“- oder „Stateful-Inspection“-Verfahren zum Einsatz. Bei diesem Verfahren schickt der Scanner ein ACK-Paket an den spezifizierten Port. Antwortet der Zielhost mit einem RST-Paket, wird der Port als „ungefiltert“ klassifiziert. Alternativ könnte das Paket von der Firewall verworfen werden oder es wird eine „ICMP Destination unreachable“-Fehlermeldung verschickt. Ist das der Fall, wird der Port als „gefiltert“ klassifiziert.

2.2.3. Fingerprinting

Beim Fingerprinting versucht der Angreifer herauszufinden, welches Betriebssystem auf dem Server verwendet wird. Hierbei kontaktiert er den Ziel-Host und kann basierend auf den Antworten (evtl. erst durch einen Abgleich mit einer Datenbank) auf das verwendete Betriebssystem schließen. Die beim Fingerprinting benutzten Anfragen können - ohne dass hier darauf eingegangen wird - nach den benutzten Pakettypen untergliedert werden. Abhängig von der Konfiguration des Ziel-Hosts kann der Hacker in kürzester Zeit sehr viele Informationen über das Betriebsystem, Patchlevel, installierte Anwendungen aufnehmen und verarbeiten. Diese Daten sind für das Identifizieren von Schwachstellen von großer Bedeutung.

2.2.4. System-Hacking

Wenn die Vorbereitungsphase abgeschlossen ist, haben Hacker eine ziemlich genaue Vorstellung vom Zielnetzwerk inklusive Hostnamen, IP-Adressen, installierte Betriebssysteme und Anwendungen. Aus diesen Informationen lässt sich eine Angriffsstrategie ableiten, die zum Beispiel bekannte Sicherheitslücken oder Fehler in der Host-Konfiguration ausnutzt, und anschließend in die Tat umsetzen.

2.2.5. Hintertüren einrichten

Meistens richten Hacker auf den kompromittierten Systemen sogenannte Backdoor-Prozesse (Hintertüren) ein, die es ihnen gestatten, zukünftig ungehindert und unbemerkt Zugang zu den Systemen zu erlangen. In diesem Zusammenhang setzen die Angreifer regelmäßig auf die Installation von Rootkits, die verschiedene ausführbare Dateien ersetzen, welche die normalen Verbindungsprozesse ausführen.

2.2.6. Spuren verwischen

Nachdem der Zutritt zum System über eine Hintertür sichergestellt ist, wird der Angreifer versuchen seine Spuren zu verwischen und die nicht mehr benötigten Dateien sowie mit dem Angriff in Verbindung stehende Logeinträge zu löschen.

2.3. Gefahr durch intern agierende Täter

Auch wenn durch die permanent wachsende Vernetzung ein Großteil der Angriffe von außen kommt, sollte man die potenzielle Gefährdung durch Innentäter keineswegs unterschätzen. In diesem Zusammenhang berichtete Carnegie Mellon University in einer kürzlich veröffentlichten Studie [Uni], dass Innentäter auch in Zukunft Unternehmen vor große Probleme stellen werden. Das Ausmaß der Bedrohung werde unterschätzt und die Vorfälle werden unzureichend aufgeklärt. Die eigene Belegschaft ist für ein Drittel aller E-Straftaten verantwortlich - so die Studie. Die Motive für kriminelle Handlungen sind unterschiedlich. Manche Angestellte identifizieren sich augenscheinlich nur unzureichend mit ihrem Arbeitgeber oder wollen damit ihrer Unzufriedenheit am Arbeitsplatz Ausdruck verleihen. Angeblich soll Letzteres auf 37 Prozent der Fälle zutreffen.

Zahlreiche Studien belegen ([Ver], [Sofa], [IDC]), dass die Innentäter grundsätzlich unterschätzt werden. Ein genauer Blick auf die Insider-Problematik zeigt, dass die meisten Straftaten den Benutzern mit weitreichenden Privilegien zugeschrieben werden (insbesondere unter den IT-Administratoren zu finden). Zu bedenken ist auch, dass die Innentäter umfassende detaillierte Kenntnisse über die IT-Infrastruktur haben und meistens auch einen (physischen) Zugang zu den Systemen haben. Eine weit verbreitete Ansicht ist, dass die Innentäter (verknüpft mit ihrer hohen Qualifikation) zwangsläufig gefährlicher seien [Sch]. Grundsätzlich gilt: Je mehr die Angreifer über das Unternehmen oder das Netzwerk wissen, desto leichter oder wirkungsvoller ist die Attacke.

„Missbrauch der Zugriffsrechte“ umschreibt ein wesentliches Problem: die Innentäter nutzen diffuse Rollenverteilungen oder großzügige Berechtigungen aus. Aus diesem Grund sind die Monitoring-Systeme in der Regel nicht in der Lage verdächtige Aktivitäten rechtzeitig zu erfassen. Eine andere weit verbreitete Fehleinschätzung betrifft den Schutz der eigenen Netze - obwohl die Notwendigkeit zur Verschlüsselung oft genug erwähnt wird, werden die internen Netze nur in den seltensten Fällen kodiert.

3. Security-Incident-Response-Prozess am LRZ

Daten zählen heutzutage zu den sensibelsten Gütern. Zum einen werden sie immer wichtiger, zum anderen wächst aufgrund neuer IT-Technologien und Verfahren der Datenbestand weiterhin exponentiell. Unabhängig davon, ob es sich um personenbezogene Daten von Mitarbeitern oder um Matrikelnummern der Studenten handelt, das Leibniz-Rechenzentrum muss zum Schutz der eigenen Reputation und im Sinne der Kostenoptimierung stets den sicheren und regelkonformen Umgang mit den gespeicherten Daten gewährleisten.

Der Einsatz von redundanten Firewall- und Intrusion-Detection-Systemen ist ein wichtiger Grundstein für die Aufrechterhaltung der Sicherheit und schützt das lokale Netz bei Angriffen von außen. Treten dennoch Sicherheitsvorfälle im internen Netz auf, würden diese Maßnahmen aufgrund ihrer Funktionsweise zu kurz greifen - Rechner der Studenten könnten beispielsweise gekapert, in Bot-Netze integriert und für das Verteilen von Malware verwendet werden. Deswegen setzt das LRZ auch strukturiert ablaufende detektierende und reaktive Maßnahmen für die Abwehr von aktuellen Bedrohungen ein, um die Vorfälle rechtzeitig zu entdecken, sie aufzuklären bzw. ihre Auswirkungen zu mildern.

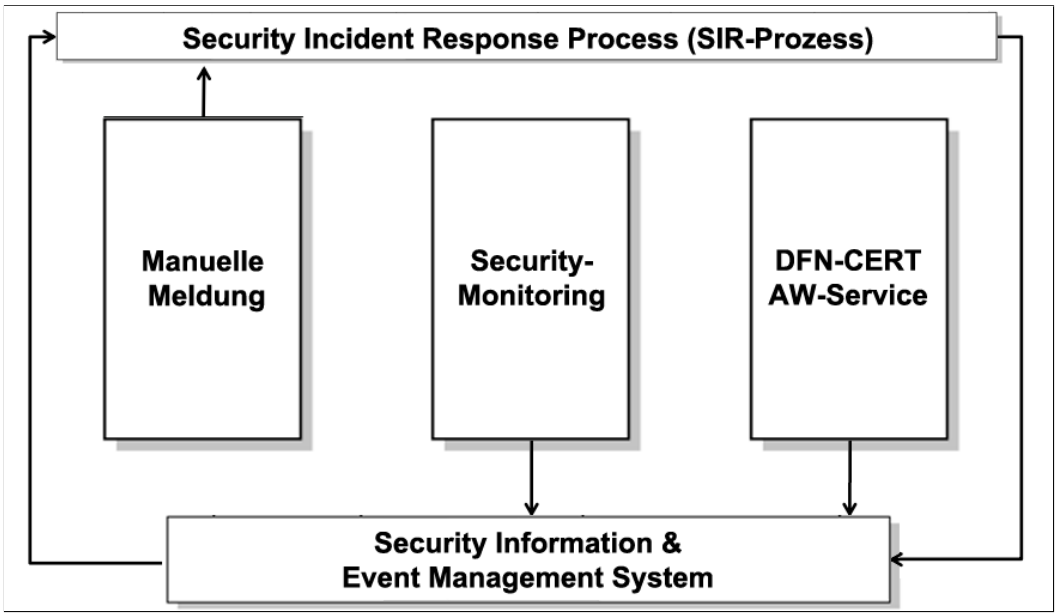

Abbildung 3.1.: Integriertes Management von Sicherheitsvorfällen am LRZ

In diesem Kapitel werden die einzelnen Bausteine erklärt aus denen sich das integrierte Management von Sicherheitsvorfällen am LRZ zusammensetzt [vgl. Abb. 3.1]. Dazu gehören unter anderem das Tool-basiertes Security Monitoring (enthält u.a. netzbasierte Intrusion Detection Systeme) und ein Security Information & Event Management (kurz SIEM). Anschließend werden die Zusammensetzung des Incident Response Teams erläutert, der am LRZ erfolgreich eingeführte, formale Security Incident Response Prozess vorgestellt und die Überschneidungen mit der IT-Forensik aufgezeigt.

3.1. Tool-gestütztes Security Monitoring

Der Schutz der eigenen IT-Infrastruktur ist auch für das Leibniz-Rechenzentrum eine zentrale Aufgabe. Vor deren Nicht-Erfüllung kann unter Umständen der Imageschaden für das Rechenzentrum immens sein. Der Zugriff auf die Systeme findet zu jeder Tageszeit statt, so dass Eingriffe Dritter, die zu einem Systemausfall führen, einen erheblichen Produktivitätsverlust verursachen können.

Gleichzeitig ist das LRZ für den Betrieb des Münchner Wissenschaftsnetz (MWN) verantwortlich und agiert als Bindeglied zwischen zentralen Server-Systemen und dezentralen Systemen in den Münchner Hochschulen und anderen wissenschaftlichen Einrichtungen [LRZb]. Die Herausforderung hierbei liegt in der dezentralen Organisation der Netzstruktur und der großen Anzahl an angeschlossenen Endgeräten. Trotzdem versucht das LRZ unerlaubte Netzzugriffe zu unterbinden bzw. bereits kompromitierte Systeme zu isolieren sowie bereits entstandenen Schaden zu minimieren. Grundsätzlich kommen folgende Abwehrmaßnahmen zum Einsatz (vgl. auch [HR10]):

- Netzbasiertes Intrusion Detection System (IDS) am Internet-Übergang (X-WiN)
- NAT-Gateway NAT-o-MAT
- E-Mail-Monitoring mittels Accounting
- Auswertung von NetFlow-Daten durch NfSen
- OSSIM - Security Information & Event Management

3.1.1. Netzbasiertes Intrusion Detection System

Die steigende Komplexität und Durchdringung der Netzwerke mit konvergenten Anwendungen (z.B. VoIP) machen die Aufrechterhaltung der Sicherheit immer aufwendiger. Das Netz muss vor Bedrohungen und unberechtigten Zugriffen sowohl von außen als auch von Innen geschützt werden. Deshalb wird der Einsatz der netzwerkbasierten Intrusion-Detection-Systeme (IDS) immer wichtiger. Beim Leibniz-Rechenzentrum kommt das Tool SNORT, ein sehr populäres und vielfältiges IDS-System, zum Einsatz.

Im Allgemeinen erledigt SNORT seine Arbeit in drei Schritten, um frühzeitig Angriffsmuster zu erkennen.: Zunächst werden Daten gesammelt, anschließend diese Daten vom System analysiert und mit bekannten Pattern verglichen und schließlich werden erkannte Regelverletzungen protokolliert.

Ein Aspekt, der für SNORT spricht, betrifft die freie Verfügbarkeit der Software sowie die sehr große Community, die dafür sorgt, dass neue Signaturen nur wenige Stunden nach Bekanntwerden eines neuartigen Angriffsmusters verfügbar sind. Zudem werden die Signaturen der SNORT-Community mit selbstentwickelten Regelsätzen methodisch erweitert, die helfen, das System für die LRZ-spezifischen Ereignisse anzupassen. Hier gilt es beispielsweise häufige Scanmuster etwa interne sowie externe SSH-Scans der Systeme innerhalb des LRZ oder MWN, zu erkennen [vgl. dazu 2.1]; sie können in Abhängigkeit von der Scannrichtung auf einen bevorstehenden Angriff bzw. auf eine bereits erfolgte Kompromitierung der internen Systeme hindeuten.

3.1.2. NAT-Gateway NAT-o-MAT

Um Systeme vor dem Angriff aus dem Internet zu schützen, bekommen Rechner in ausgewählten Subnetzen des MWN (Münchner Wissenschaftsnetz) eine private IP-Adresse. Um Daten zwischen Intra- und Internet austauschen zu können, müssen private/interne IP-Adressen in öffentliche/externe IP-Adressen übersetzt werden (Network Address Translation, kurz NAT). Für die Adressumwandlung wird am LRZ eine erweiterte Version eines NAT-Gateways eingesetzt, welches, zuzüglich zur Umwandlung der privaten Quell-IP-Adressen in öffentliche Quell-IP-Adressen, gleichzeitig auch Aufgaben eines Security-Gateways wahrnimmt und Funktionen zur Bandbreitenregelung sowie der Verhinderung von Portscans und Denial of Service liefert. Als Folge können externe Systeme von sich aus keine Verbindung mit den Hosts im Intranet aufbauen.

Für die Absicherung der Systeme wird ein regelbasiertes Modell angewendet [vgl. auch [LRZa]]. Wird eine der festgelegten Regeln verletzt, z.B. beim Abschicken von mehr als 10 Paketen pro Sekunde, erhält der Rechner einen Strafpunkt. Bei Überschreiten der festgelegten Strafpunktegrenze innerhalb von fünfzehn Minuten wird davon ausgegangen, dass das System möglicherweise infiziert ist und für ein Zeitfenster von fünfzehn Minuten blockiert. Das System wird wieder freigeschaltet, sofern die Grenze von 120 Strafpunkten innerhalb des besagten Zeitfensters wieder unterschritten wird.

Nachfolgend wird eine exemplarische Auswahl der Analyseergebnisse des NAT-o-MAT-Gateways aufgeführt.

```
Gesperrt seit / Blocked since 15.11.10 05:20
Überschreitungen   Protokoll Zielport und Grund der Sperrung
Number of hits     Protocol  Destination port and suspension reason
111                UDP       4600-4699 - Edonkey Filesharing
30                 UDP       7560
30                 UDP       5514
30                 UDP       54067
30                 UDP       4600-4699 - Edonkey Filesharing
30                 UDP       43793
30                 UDP       37283
```

Abbildung 3.2.: NAT-o-MAT - NAT-Gateway und Security-Monitoring.

Quelle: [LRZa, S. 7]

3.1.3. E-Mail-Monitoring mittels Accounting

Am LRZ wird ein E-Mail-Monitoring-Mechanismus als Hilfsmittel bei der Spam-Abwehr genutzt: das Verhalten der mailversendenden Rechner wird überwacht und ausgewertet. Dadurch lassen sich schnell und gezielt Systeme identifizieren, die eine sehr große Anzahl an E-Mails gleichzeitig versenden. Es wird i.d.R. davon ausgegangen, dass sie mit Schadsoftware infiziert und unter Umständen in ein Botnetz integriert wurden.

Im Gegensatz zu geläufigen Anti-Spam-Lösungen, die beispielsweise simple Bayessche Regeln für die Inhaltsanalyse der ein- und ausgehenden Nachrichten nutzen sowie die Identität und Reputation des Absenders überprüfen, überwacht die vom Leibniz-Rechenzentrum eingesetzte Lösung den E-Mail-Verkehr und zählt jede Mailverbindung in zwei fest definierten Zeitintervallen (5 sowie 60 Minuten) und dies über heterogene Umgebungen hinweg. Beim Überschreiten eines individuell festgelegten Schwellenwerts wird eine Warnung an den verantwortlichen Administrator geschickt. Außerdem erlaubt die Implementierung das betroffene System direkt manuell zu sperren, falls dies erforderlich ist. Ein Beispiel für solch einen Fall ist der Abbildung 3.3 zu entnehmen.

```
Monitoring-Details
==================

Rechner:                [xxx.xxx.xxx.xxx]

Monitoring-Intervall:   5 Minuten  vor  20.07.2010  10:30:28
Mail-Verbindungen:      1616
```

Abbildung 3.3.: E-Mail-Monitoring mittels Accounting.

Quelle: [LRZa, S. 8]

Zu den weiteren Vorteilen der Lösung gehören die Performanz und die geringe Wahrscheinlichkeit des Fehlalarms. IT-Administratoren können darüber hinaus Ausnahmen für die Mailserver nach Bedarf definieren, die täglich ein sehr hohes Mailaufkommen haben.

3.1.4. Auswertung von NetFlow-Daten durch NfSen

Veränderungen in den Angriffstechniken stellen IT-Administratoren am LRZ vor immer neue Herausforderungen. Von zentraler Bedeutung ist dabei unter anderem die Analyse von NetFlow-Daten zum Zweck der Erkennung von unbekannten Angriffstypen. Daneben spielt der Einsatz dieser Auswertemechanismen eine zentrale Rolle bei der Erkennung von Traffic-Anomalien. Vertiefende Untersuchungen mittels NetFlow-Analyse sind als Ergänzug für das netzbasiertes Intrusion Detection System SNORT anzusehen.

Netflow ist eine leistungsfähige Technologie, die von der Firma Cisco entwickelt wurde und heute hauptsächlich in vielen Layer-3 Geräten wie Routern und Switches in verschiedenen Versionen zu finden ist. Zentrale Aufgabe von NetFlow ist der Export der gesammelten Informationen über den Datenstrom zu einem vorhanden NetFlow-Collector. Jeglicher Verkehrsfluss wird erfasst und in periodischen Abständen „aggregiert“ zum NetFlow-Collector geschickt. Die gesendeten Informationen beinhalten im Einzelnen:

- Protokollnummer
- Quell-IP-Adresse
- Ziel-IP-Adresse
- Quell-Port-Nummer
- Ziel-Port-Nummer
- Type of Service
- Input Interface

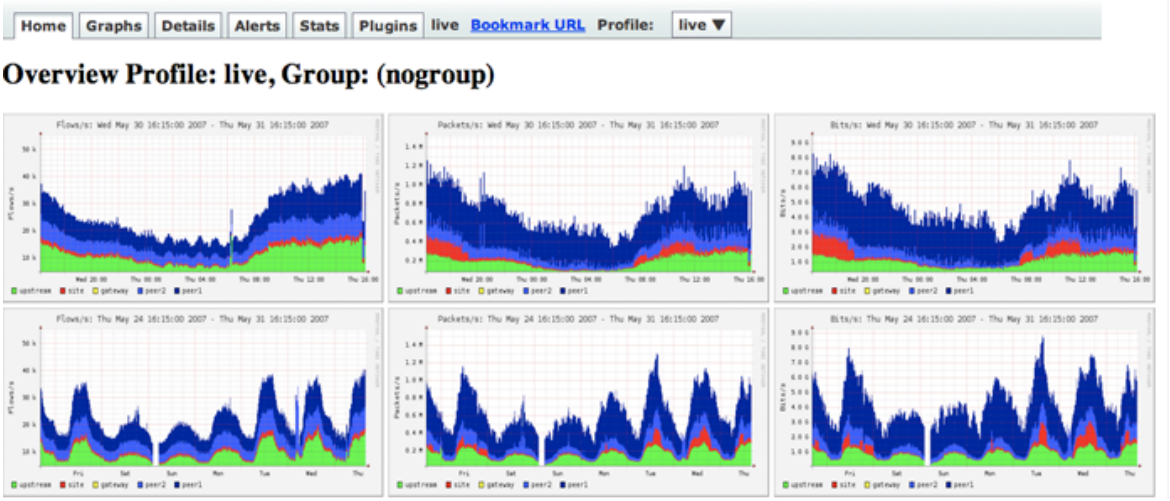

Abbildung 3.4.: Integrierte Dashboards in NfSen.

Quelle: [NfS]

Leider fallen bei NetFlow-Daten sehr schnell sehr große Datenvolumen an. Um sich trotzdem einen Überblick verschaffen zu können, kommt am Leibniz-Rechenzentrum das quelloffene Web-Frontend Netflow Sensor (NFSen) zum Einsatz. NFSen erlaubt grundsätzlich die Aufteilung der über NetFlow erfassten Informationen nach Protokollen sowie Sortierung des Traffics nach Flow, Paketen sowie übertragenem Volumen. Das System bietet Unterstützung für benutzerdefinierte Profile und ein umfangreiches Live-Reporting.

3.1.5. Security Information & Event Management

Die Open Source Security Information Management Plattform (OSSIM) ist ein äußerst mächtiges Werkzeug für Aufgaben im Bereich der zentralen Sammlung, Korrelation und Auswertung von sicherheitsrelevanten Ereignissen. Die Software wird durch die Firma AlienVault aktiv weiterentwickelt.

Ein einheitliches Erfassen und Darstellen der sicherheitsrelevanten Ereignisse ist in der neuesten Version (3.1) noch effizienter: Die am Leibniz-Rechenzentrum zum Einsatz kommenden Systeme für die Überwachung der Netzwerk- und Systemsicherheit werden bei OSSIM über ein Framework zentral angebunden. Durch die Korrelation wichtiger Ereignisse aus verschiedenen Quellen können IT-Administratoren schnell und gezielt auf aktuelle Bedrohungen reagieren und unautorisierte Zugriffe verhindern. Für die Korrelation stehen unterschiedliche Mechanismen zur Verfügung. Diese beinhalten im Einzelnen:

- Verknüpfung mehrerer Ereignisse
- Definition von Schwellwerten und Alarmierung bei einer Überschreitung
- Abfragen auf den Inhalt von Datenfeldern
- Durchführung von Aktionen

Es lassen sich automatisierte Aktionen festlegen: IT-Administratoren können von der Management-Konsole aus benutzerdefinierte Policies anlegen und optimieren, um auf bestimmte Ereignisse, die sich aus dem eigentlichen Anwendungsfall ergeben, adäquat reagieren zu können. Das Spektrum möglicher Reaktionen umfasst eine Vielzahl von Möglichkeiten: angefangen bei der Benachrichtigung im Web-Frontend bis hin zum Ausführen eines externen Programms. Zu den Vorteilen des automatisierten Ereignismanagements gehört, dass Risiken für Ausfälle minimiert und der Betrieb der IT-Infrastruktur erleichtert werden.

3.2. DFN-CERT Sicherheitsmeldungen

Der DFN-CERT bietet seinen Anwendern eine Reihe von reaktiven und präventiven Dienstleistungen an, um das Eintreten von Sicherheitsvorfällen zu verhindern bzw. Auswirkungen von Vorfällen abzumildern und deren Beseitigung zu unterstützen. Informationen über sicherheitsrelevante Ereignisse ist eine Kernvoraussetzung für eine effiziente Absicherung der IT-Infrastruktur. Gebündelt werden alle sicherheitsrelevanten Dienste im DFN-CERT Portal. Der Zugang zu den einzelnen Diensten ist zertifikatgeschützt. Das gleiche gilt auch für den Zugriff auf die Konfigurationsdateien. Folgende Dienste werden durch das DFN-CERT angeboten:

- Automatische Warnmeldungen
- Informationen zu Schwachstellen
- Netzprüfer (derzeit im Testbetrieb)

3.2.1. Automatische Warnmeldungen

Cyber-Angriffe führen immer wieder zum Erfolg. Nur rechtzeitige Erkennung ermöglicht es IT-Administratoren, notwendige Gegenmaßnahmen zu ergreifen, um den Schaden einzudämmen. Genau an dieser Stelle setzt der Dienst Automatische Warnmeldungen des DFN-CERT an: werden Auffälligkeiten im Zusammenhang mit den IP-Adressen des LRZ bzw. des MWN festgestellt, werden einmal täglich automatisch generierte E-Mails mit Warnmeldungen versandt. Die Grundlage bilden zum einen Auswertung und Analyse von Daten aus verschiedenen öffentlichen Quellen und zum anderen der Betrieb der eigenen Sensoren, sog. Darknets sowie Honeypots.

Darknets sind Netzblöcke, die keine Dienste anbieten und daher im Regelfall keine Verbindungen annehmen bzw. aufbauen. Werden auffällige Scan-Versuche in einem bestimmten Portbereich festgestellt (z.B. Suche nach aktiven Webservern mit veralteter phpMyAdmin-Installation [siehe auch 2.1]), so ist es ein Hinweis auf ein mögliches Sicherheitsproblem.

Honeypots sind emulierte Dienste, die zur Erkennung, Untersuchung und Verfolgung der Einbrüche eingesetzt werden. Jedes Mal, wenn der Honeypot eine Verbindung entgegennimmt oder sogar selbst eine Verbindung aufbaut, kann man von einem Einbruch ausgehen. Wird ein Honeypot angegriffen, überträgt der Angreifer die Schadsoftware an den Server. Der Angriff selbst wird protokolliert und die Malware von verschiedenen Scannern untersucht.

Anschließend werden die gesammelten Daten korreliert sowie normiert und allen DFN-Anwender durch automatisch generierte E-Mails zur Verfügung gestellt. Einen Auszug aus einer AW-Nachricht sieht folgendermaßen aus (siehe Abbildung 3.5).

```
System:       xxx.xxx.xxx.xxx
Meldungstyp:  Bot
Zeitstempel:  2010-07-17 09:31:12 GMT+0200 (Sommerzeit)
Beschreibung: Auf dem System scheint eine Bot-Software betrieben zu
              werden, die versucht, einen HTTP-basierten Bot-Netz
              Control-Server zu erreichen. Zu den unterschiedlichen
              Malwaretypen finden Sie unter der folgender Webseite mehr
              Informationen: http://www.cert.dfn.de/index.php?id=bot

TCP Quellport   TCP Zielport   Malwaretyp    HTTP Request
-------------------------------------------------------------------
1667            80            Conficker     GET /search?q=0 HTTP/1.0
3812            80            Conficker     GET /search?q=0 HTTP/1.0
```

Abbildung 3.5.: Auszug aus einer durch den DFN-CERT Dienst Automatische Warnmeldungen verschickten E-Mail.

Quelle: [LRZa, S. 4]

3.2.2. Informationen zu Schwachstellen

Neben dem Dienst Automatischen Warnmeldungen besteht die Möglichkeit, sich über das DFN-CERT Portal über mögliche Schwachstellen in den Systemen informieren lassen, die am LRZ eingesetzt werden. Nur wenn IT-Administratoren Kenntnis über Schwachstellen haben, können sie angemessene Gegenmaßnamen treffen. Alleine im Jahr 2010 wurden mehr als 6000 neue Schwachstellenmeldungen veröffentlicht [[Fos10]]. Meldungen zu Schwachstellen enthalten Informationen zu den betroffenen Plattformen und eine Erklärung, wie die Schwachstelle von einem Angreifer ausgenutzt werden kann. Ferner lassen sich den Einträgen folgende Informationen entnehmen [[GF09], S.41]:

- Unter welchen Voraussetzungen ist ein System betroffen?
- Wie können Angriffe auf die Schwachstelle z. B. in den Log-Daten des Systems erkannt werden?
- Wie kritisch ist die Schwachstelle und wird sie bereits in der Praxis ausgenutzt?
- Steht für das betroffene Programm ein Update bereit oder gibt es einen temporären Workaround?

Durch diese Erkenntnisse lassen sich zeitnah konkrete Schritte seitens der verantwortlichen Administratoren einleiten und die Erfolgschancen für bestimmte Cyber-Angriffsarten deutlich reduzieren. So werden schon im Vorfeld mögliche Schäden vermieden.

3.2.3. „Netzwerkprüfer“

Neben den Informationen zu Schwachstellen und dem AW-Dienst steht den DFN-Nutzern ein unverzichtbares Hilfsmittel für regelmäßige Vulnerability Assessments zur Verfügung - der Dienst „Netzwerkprüfer“. Dieser Dienst wurde erst vor wenigen Jahren in das DFN-CERT Portal integriert. Das Scan-Werkzeug basiert auf dem

Nmap-Scanner und ermöglicht eine Überprüfung der Erreichbarkeit der lokalen Systeme von außen, indem ein Scan von Netzbereichen oder einzelnen IP-Adressen durchgeführt wird. Die Scan-Ergebnisse der derzeit ca. 2.500 wichtigsten TCP- und UDP-Ports werden normiert und tabellarisch aufgelistet. Netzwerkprüfer bietet den LRZ-Administratoren die Möglichkeit, regelmäßige Scans automatisch durchführen zu lassen sowie Ergebnisse von zwei Scandurchläufe miteinander zu vergleichen, um theoretische Bedrohungssituation besser einschätzen zu können. Ein wichtiges Merkmal des Dienstes ist, dass anhand der Ergebnisse eines Scans Administratoren leichter Fehlkonfigurationen der Hosts erkennen und entsprechend handeln können - dies ist unverzichtbar für eine erfolgreiche IT-Sicherheitsstrategie.

3.3. Security Incident Response Team

Mit der rasanten Fortentwicklung der Informationstechnik im letzten Jahrzehnt häuften sich die Sicherheitsvorfälle am LRZ - und machten damit auch, bedingt durch die Heterogenität der IT-Infrastruktur, die Einrichtung eines dedizierten Security-Incident-Response-Teams erforderlich. Ein solches Team ist hierarchisch organisiert und beschäftigt sich in erster Linie mit der Lösung von konkreten IT-Sicherheitsvorfällen und der Erstellung sicherheitstechnischer Analysen. Um die Arbeitsweise des CSIRT so effektiv und kosteneffizient wie möglich zu machen, wurde bei der Einrichtung darauf geachtet, das Team so klein wie möglich zu halten. Die Aufgaben des LRZ-CSIRT werden derzeit von 10 Mitarbeitern erfüllt, die eine Reihe von Fähigkeiten mitbringen sollten: angefangen beim umfassenden Wissen über Hardware und Betriebssysteme bis hin zu Programmierung und IT-Sicherheit. Sie nehmen folgende Positionen wahr:

- **CSIRT-Hotliner** - Mitglied des LRZ-CSIRT, das sicherheitsrelevante Meldungen entgegennimmt sowie für die Kommunikation mit Vorfallmelder, Verantwortlichen und externen CSIRTs zuständig ist.
- **LRZ-CSIRT-Mitglied**er - Spezialisten für die Erfassung und Behandlung des Vorfalls. Sie übernehmen in der Regel unterstützende Aufgaben (z.B. Dokumentation).
- **System- oder Netzadministratoren** - Spezialisten, die für die Incident-Bearbeitung zuständig sind. Sie verfügen über weitreichende Systemkenntnisse und Zugriffsberechtigungen, um nach Hinweisen und Spuren zu suchen.
- **IT-Forensiker** - Spezialisten, die sich mit der Untersuchung von kompromittierten Systeme beschäftigen und Beweise zivil- bzw. strafrechtlich verwertbar sicherstellen.
- **Security Incident Coordinator** (SIC) - Leiter des Teams, der über fundierte Kenntnisse in bestimmten Fachbereichen verfügt, Incident-Bearbeitung plant, koordiniert und begutachtet. Er sorgt für das exakte Zusammenspiel aller Beteiligten.

Dabei können Mitglieder des CSIRT gleichzeitig mehrere dieser Funktionen erfüllen - ein Hotliner ist in der Regel auch für die Bearbeitung eines Sicherheitsvorfalls zuständig. Folgende Tätigkeiten fallen in den Aufgabenbereich des Security-Incident-Response-Teams:

- Bereitstellung einer kompetente Hotline von Montag bis Freitag zu den allgemeinen Geschäftszeiten (Ausnahmen sind gesetzliche, nationale und bayerische Feiertage, sowie der 24.12. und 31.12.),
- Reaktion auf alle sicherheitsrelevanten Zwischenfälle oder Verdachtsfälle unter besonderer Berücksichtigung der Richtlinien für einen standardisierten Untersuchungsprozess,
- Zeitnahe Bestätigung, ob es sich um ein Security Incident oder um einen Fehlalarm handelt,
- Durchführung einer unvoreingenommenen Untersuchung (bei Bedarf in Abstimmung mit Gruppen- und Abteilungsleitung, SIC und ggf. LRZ-Leitung),
- Quantifizierung des angerichteten Schadens und des Umfang eines Zwischenfalls,
- Identifizierung der ausgenutzten Sicherheitslücken bzw. der Methoden, die zur Kompromittierung geführt haben,
- Eindämmung des Vorfalls (in Abstimmung mit SIC),

- Sammlung aller relevanten Beweise im Zusammenhang mit einem Vorfall und Erstellung einer umfassenden Dokumentation,
- Nach Incident-Abschluss werden betroffene IT-Komponenten 5 Werktage durch Mitglieder des LRZ-CSIRT beobachtet, um erneute Kompromittierung auszuschließen.

Neben der reinen Incident-Bearbeitung beschäftigen sich die Fachleute des CSIRT-Teams mit diversen fortlaufenden organisatorischen Maßnahmen: Die Sicherheitsvorfälle werden analysiert und das Konzept für die Sicherheitsvorfallbehandlung bei Bedarf aktualisiert. Auf dieser Basis erfolgt die Festlegung und Überprüfung eines Security-Monitoring-, Alarmierungs-, Datensicherungs- sowie Patch- und Updatemanagementkonzepts. Konzept sowie Überprüfung hinsichtlich Vollständigkeit und Umsetzbarkeit gehen somit Hand in Hand. Das garantiert einen reibungslosen und effektiven Ablauf bei der Bearbeitung von Sicherheitsvorfällen. Weiterführende Informationen können [Met11] entnommen werden.

3.4. Security Incident Response Prozess

Schon früh wuchs am LRZ die Erkenntnis, wie wichtig ein sicheres und zuverlässiges Meldesystem für Sicherheitsvorfälle in verteilten Systemen ist - durch die immer komplexere Verflechtung der Netzstrukturen bieten sich immer wieder neue potentielle Angriffspunkte für externe Hacker an. Im Laufe der Jahre entstand ein modernes, mehrgliedriges System, welches es dem LRZ ermöglicht, für den Notfall gerüstet zu sein. Die Meldewege des SIR-Prozesses umfassen, wie in den vorherigen Abschnitten beschrieben, neben verschiedenen internen Monitoring-Mechanismen auch automatische Warnmeldungen des DFN-CERT. Zusätzlich steht den IT-Administratoren die Möglichkeit frei, manuelle Meldung per E-Mail oder Telefon an das LRZ-CSIRT zu senden.

Es existieren diverse Szenarien, die die Kontinuität von Dienstbereitstellung am LRZ beeinträchtigen können. Um beim Eintreten eines Notfalls schnell und zielgerichtet reagieren zu können, wurde Mitte 2010 ein formaler Prozess zur Bearbeitung von Security Incidents eingeführt. Eine Einteilung des möglichen Lösungsablaufs in mehrere Phasen bildet die Grundlage für eine systematische Incident-Bearbeitung beim Eintreten eines Vorfalls und (sofern zutreffend) eine zeitnahe Wiederherstellung der Verfügbarkeit der entsprechenden Ressourcen. Im Wesentlichen besteht der normale Ablauf von der Meldung eines Sicherheitsvorfalls bis hin zur Aufklärung in der Regel aus folgenden Schritten (vgl. Abb. 3.6):

- Incident-Aufnahme,
- Incident-Klassifikation & Priorisierung,
- Incident-Bearbeitung,
- Incident-Lösung,
- Incident-Abschluss.

3.4.1. Incident-Aufnahme

Bei der Meldung eines Vorfalls gibt es einige elementare Dinge zu beachten. Die Erfahrung zeigt, dass Informationen, die nicht frühzeitig erfasst werden, niemals dokumentiert werden. Für eine Untersuchung sind folgende Informationen sehr wichtig:

- aktuelle Uhrzeit,
- Kontaktinformationen des Vorfallmelders,
- Wer oder welches System meldete den IT-Sicherheitsvorfall,
- Art und Typ des Vorfalls,
- vermuteter Zeitpunkt des Vorfalls,
- vorliegende Informationen über einen potenziellen Angreifer,

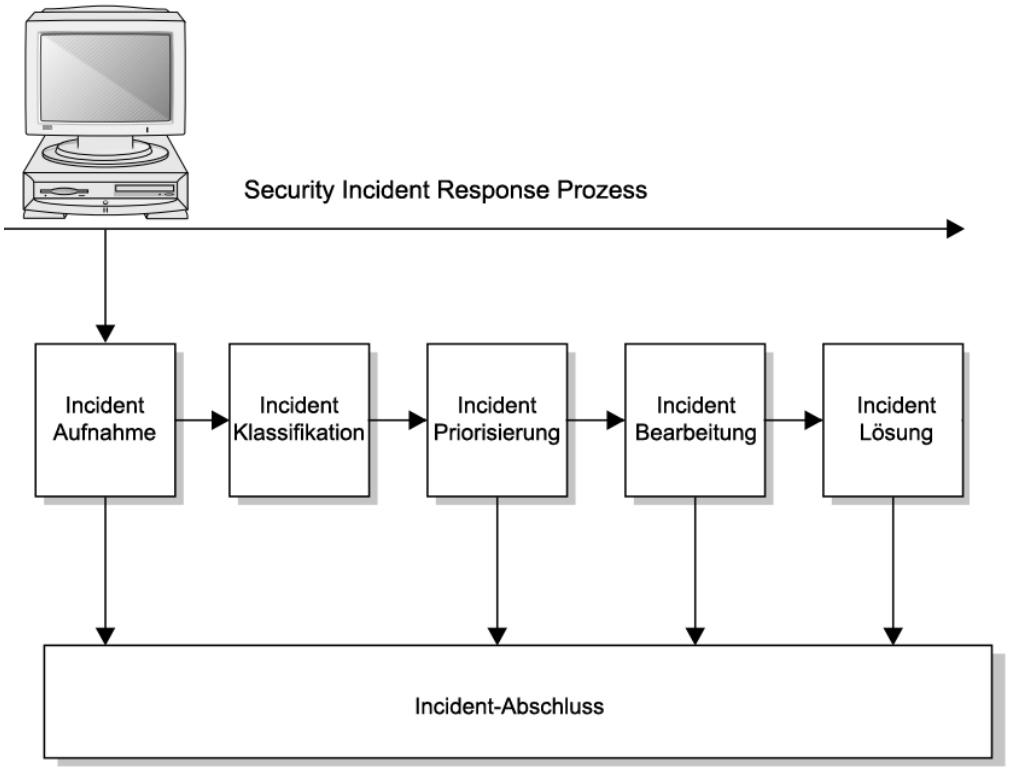

Abbildung 3.6.: Security Incident Response Prozess am LRZ

- Standort der betroffenen Systeme,
- nähere Informationen über Hardware, Betriebssystem und installierte Software,
- weiterführende Informationen über das System,
  - Name des zuständigen Systemadministrators,
  - Wichtigkeit der gestohlenen oder beschädigten Daten,
  - Netzwerkname und IP-Adresse der betroffenen Systeme.
- Eine Auflistung der Personen, die seit dem Bekanntwerden des Vorfalls Zugang zu den Systemen hatten,
- bereits unternommene Abwehrmaßnahmen.

Die Incident-Aufnahme ist ein methodischer Prozess. Daher hat jeder LRZ-Administrator ein Merkblatt griffbereit, das die zuvor kurz erwähnten Fragestellungen sowie Empfehlungen für Verhaltensmaßnahmen im Krisenfall enthält. Dieser Ansatz macht es möglich, frühzeitig die richtigen Sachverhalte zu erfragen.

3.4.2. Incident-Klassifikation & Priorisierung

Grundlegend wichtig ist eine klare Klassifizierung des Vorfalls - sämtliche zur Verfügung stehende Informationen werden analysiert und auf Vollständigkeit geprüft. Beispielsweise gehören dazu eine Einschätzung der Fähigkeiten und der Herkunft des Angreifers sowie eine Bestimmung des Standorts und der Anzahl der betroffenen Systeme. Es gilt herauszufinden, um welche Angriffsart es sich vermutlich gehandelt hat und welche Daten theoretisch hätten eingesehen werden können. Aus diesen Erstdaten wird die genauere Einschätzung des Schadens sowie eine Einteilung in eine von vier Prioritätsklassen möglich. Eine frühzeitige Klassifizierung des Vorfalls ist unabdingbar: Sie erleichtert die Wahl der passenden Abläufe. Unbedachte Reaktionen werden so

bereits im Ansatz verhindert. Aufgrund der Vielzahl der Angriffe werden für Routinevorfälle Richtlinien definiert, so dass Standard Security Incidents effektiver gehandhabt werden könnten.

3.4.3. Incident-Bearbeitung

Die Erfassung der vorfallsrelevanten Daten sowie Klassifizierung - und insbesondere Priorisierung - des Vorfalls sind notwendig, um die richtige Response Strategie auszuwählen. Es ist entscheidend, die Auswirkungen der Maßnahmen richtig abzuschätzen und die richtigen Abläufe zu starten. Ebenfalls stehen dem LRZ-CSIRT Schritt-für-Schritt Anleitungen für Routinetätigkeiten zur Verfügung, da dadurch eine Zeitreduktion bei der Vorfallsbearbeitung erzielt werden kann - die Abläufe werden durch vorher definierte Prozesse effizient gesteuert. Leider ist es nicht möglich, für jeden auftretenden Fall ein maßgeschneidertes detailliertes Konzept zu haben.

Je nach Priorisierung der Dienstverfügbarkeit, Minimierung der Downtime bzw. lückenlosen Auflärung des Vorfalls werden von den Mitgliedern des LRZ-CSIRT Response Strategien ausgewählt, die die jeweils spezifischen Gegebenheiten berücksichtigen. Bei der Auswahl einer Response Strategie werden folgende Punkte beachtet:

- Wie kritisch sind die betroffenen Systeme für die Abläufe am LRZ?
- Wieviel Zeit muss für die Abwehrmaßnahmen eingeplant werden?
- Wie wichtig sind die auf dem System bereitgestellten Daten?
- Müssen rechtliche Aspekte berücksichtigt werden?
- Ist der Vorfall bereits an die Öffentlichkeit gelangt?
- Welche Störungen, Schäden und Folgeschäden ergeben sich konkret zu diesem Zeitpunkt?
- Welche Kosten wird der Sicherheitsvorfall bzw. dessen Aufklärung verursachen?

Unabhängig davon liegt das Hauptaugenmerk für alle Schadensszenarien auf der Erarbeitung geeigneter Vorgehensweisen für die schnellstmögliche Lösung des Vorfalls und die Wiederaufnahme des Produktivbetriebs.

3.4.4. Lösung und Abschluss des Incidents

Nach der Durchführung aller Reaktionsmaßnahmen wird untersucht, ob die betroffenen Systeme irgendwelche Auffälligkeiten zeigen und ob weitere Maßnahmen notwendig sind. Weiterhin wird überpüft, ob die Sicherheitslücken geschlossen wurden und hinterlassene Manipulationen wie trojanische Pferde wirklich beseitigt wurden. Sollte dies nicht der Fall sein, müssen weitere Abwehrmaßnahmen ergriffen werden, ansonsten kann der Vorfall als vorläufig gelöst betrachtet werden. Sind innerhalb von 14 Werktagen nach Rückkehr zum Regelbetrieb keine weiteren Auffälligkeiten beobachtet worden, gilt der Sicherheitsvorfall LRZ-seitig als abgeschlossen [Met11].

Ein weiterer Aspekt ist die Dokumentation des gesamten SIR-Prozesses. Hierfür werden alle Aktionen möglichst detailliert dokumentiert. Diese Vorgehensweise erlaubt den Mitgliedern des CSIRT sich einen Überblick über Ursachen, Auswirkungen und Maßnahmen zu verschaffen und die aufgetretenen Probleme nachvollziehbar zu machen. Die Dokumentation dient somit als Template für die Ausarbeitung von neuen Routinen. Somit sind Anpassungen der Reaktionsstrategien einfach realisierbar.

4. Einführung in die Computer-Forensik

In diesem Kapitel werden die wichtigen Sachverhalte rund um IT-Forensik erklärt sowie die konkreten Anforderungen an den Ermittlungsprozess erläutert. Der letzte Abschnitt beschreibt die einzelnen Phasen der Ermittlung. Die Kenntnis dieser Sachverhalte ist eine der Voraussetzungen für die korrekte Durchführung einer computer-forensischen Analyse.

4.1. Ziele

Das Thema der Computer-Forensik ist sehr umfassend und berührt viele unterschiedliche technische Bereiche. Im Rahmen einer IT-Forensik-Untersuchung verfolgen die Ermittler in der Regel folgende Ziele:

- Quantifizieren des entstandenen Schadens nach einem Systemeinbruch,
- Identifizierung der verantwortlichen Sicherheitslücken,
- Identifikation des Angreifers,
- Erfassung, Analyse und Auswertung digitaler Spuren für weitere juristische Aktionen.

Im Zentrum der Ermittlung steht dabei ein Komplex aus sechs essentiellen Fragestellungen bezüglich der Ereignisse, die zur Untersuchung geführt haben:

- Was ist passiert?
- Wo ist es geschehen?
- Wann ist es passiert?
- Wie ist es passiert?
- Wer hat es getan?
- Was kann gegen eine Wiederholung getan werden?

Um diese Fragen beantworten zu können, wird zunächst eine umfangreiche Datenspurenakquisition durchgeführt. Angesichts der Komplexität des Ermittlungsprozesses sind sechs Faktoren gefordert, deren Beachtung bei der Durchführung einer computer-forensischen Analyse die allerhöchste Wichtigkeit beigemessen werden soll. Sie werden im folgenden Abschnitt einzeln vorgestellt.

4.2. Anforderungen an den Ermittlungsprozess

Für die Erfolgsmessung des Ermittlungsprozesses haben sich in den letzten Jahren eine Reihe von Kriterien etabliert (vgl. [Ges10]). Eines der wichtigsten Kriterien - von der auch viele andere abhängen - ist die **Akzeptanz** (AK). Die angewandten Methoden und Schritte müssen in der Fachwelt beschrieben und allgemein akzeptiert sein.

Davon abgeleitet ist die **Glaubwürdigkeit** (GW). Die Robustheit und Funktionalität von Methoden wird vorausgesetzt und muss ggf. plausibel nachgewiesen werden.

Ein weiterer Punkt ist die Voraussetzung der **Wiederholbarkeit** (WH) der Ergebnisse. So gilt es sicherzustellen, dass bei Anwendung der Methoden durch Dritte die gleichen Ergebnisse produziert werden.

Auch die **Integrität** (IN) der Beweisspuren gehört zur Durchführung einer erfolgreichen Ermittlung [5.9.2]. Es muss jederzeit belegt werden können, dass die sichergestellten Spuren im Rahmen der Ermittlung nicht unbemerkt verändert wurden.

Ein weiterer Aspekt, den IT-Forensik vorgibt, betrifft die Herleitung der logisch nachvollziehbaren Verbindungen zwischen Personen, Ereignissen und Beweisspuren. Man spricht in diesem Zusammenhang von **Ursache und Auswirkungen** (UA). Hierfür existieren methodische Vorgehen, die helfen, die Zusammenhänge herzustellen.

Bei der Fallbearbeitung soll ferner für jeden Schritt des Ermittlungsprozesses eine angemessene **Dokumentation** (DO) angelegt werden. Erfahrungsgemäß werden Informationen und Tätigkeiten, die nicht sofort dokumentiert werden, niemals erfasst werden. Die Protokollierung sollte so erfolgen, dass sie später vor Gericht verwertbar ist.

Abgesehen von diesen essentiellen Anforderungen versucht der vorliegende Leitfaden, eine **Reduktion der Einarbeitungszeit** (RE) für die Mitarbeiter des Leibniz-Rechenzentrums zu erreichen. Die Erkenntnisse aus diesem Leitfaden sollen es ermöglichen, forensisch korrekte Vorgehensweisen anzuwenden und eine Lektüre einschlägiger Publikationen überflüssig machen.

Die **Verfügbarkeitsanforderungen** (VA) der LRZ-Systeme können von Hochverfügbarkeitsanforderungen mit kurzen Wiederanlaufszeiten bis hin zu Systemen reichen, bei denen mehrere Tage bis zur Wiederinbetriebnahme tolerierbar sind. In Abhängigkeit dazu sind unterschiedliche Vorgehensweisen erforderlich, das Reduzieren der forensischen Schritte auf ein absolutes Minimum ist denkbar - sofern die Verfügbarkeitsanforderungen als sehr hoch einzustufen sind (vgl. dazu 4.4).

Insgesamt muss der **Integration** (IG) des Forensik-Leitfadens in die bestehenden Strukturen Rechnung getragen werden, da unter anderem Abläufe für Reaktion auf Sicherheitsvorfälle bereits etabliert und im praktischen Einsatz verifiziert wurden. Ziel dieses Leitfadens ist eine aktive Unterstützung bei der Incident Response beim gleichzeitigen hohen Interaktionsgrad mit dem LRZ-SIR-Prozess.

4.3. Phasen der Ermittlung

An dieser Stelle sei kurz auf den allgemeinen Ablauf einer forensischen Untersuchung eingegangen. Die Vorgehensweise wird in logisch zusammengehörige Abschnitte untergegliedert und basiert - wie bereits angesprochen - auf dem LRZ-SIR-Prozess. Es werden alle Aspekte betrachtet, die zur erfolgreichen Aufklärung erforderlich sind. Dieser Leitfaden beinhaltet eine Aufteilung in sechs Abschnitte zur Bewältigung der Untersuchung:

- Entdecken eines Sicherheitsvorfalls.
- Erste schnelle Sammlung von Spuren.
- Entscheidung über weiteres Vorgehen.
- Elektronische Beweise werden vollständig sichergestellt.
- Beweisspuren werden identifiziert und analysiert.
- Analyseergebnisse werden in einem Abschlussbericht zusammengefasst.

Der Ausgangspunkt für eine forensische Untersuchung ist - vergleichbar mit dem SIR-Prozess - die Erkennung der Anzeichen für einen Angriff. Dazu ist es erforderlich, dass die Mitarbeiter des LRZ-CSIRT-Teams ausreichend Informationen über den aktuellen Zustand der IT-Systeme und sonstige Ressourcen haben. Diese Informationen können beispielsweise aus netzseitigen Überwachungssystemen, Auswertung serverseitiger Hinweise und externer Meldungen kommen. Nur so lassen sich Kompromitierungen der IT-Infrastruktur rechtzeitig erkennen.

Die zweite wichtige Phase besteht - in Übereinstimmung mit dem SIR-Prozess - aus der ersten schnellen Sammlung von Spuren, damit eine angemessene Response-Strategie gewählt werden kann. Neben der effektiven Eindämmung der aus dem Sicherheitsvorfall resultierenden Schäden, wird eine große Bedeutung der

Aufklärung der Ursachen für den erfolgreichen Einbruch beigemessen - es wird sorgfältig abgewogen, ob die Aufklärung des Schadens gegenüber der Eindämmung Vorrang hat.

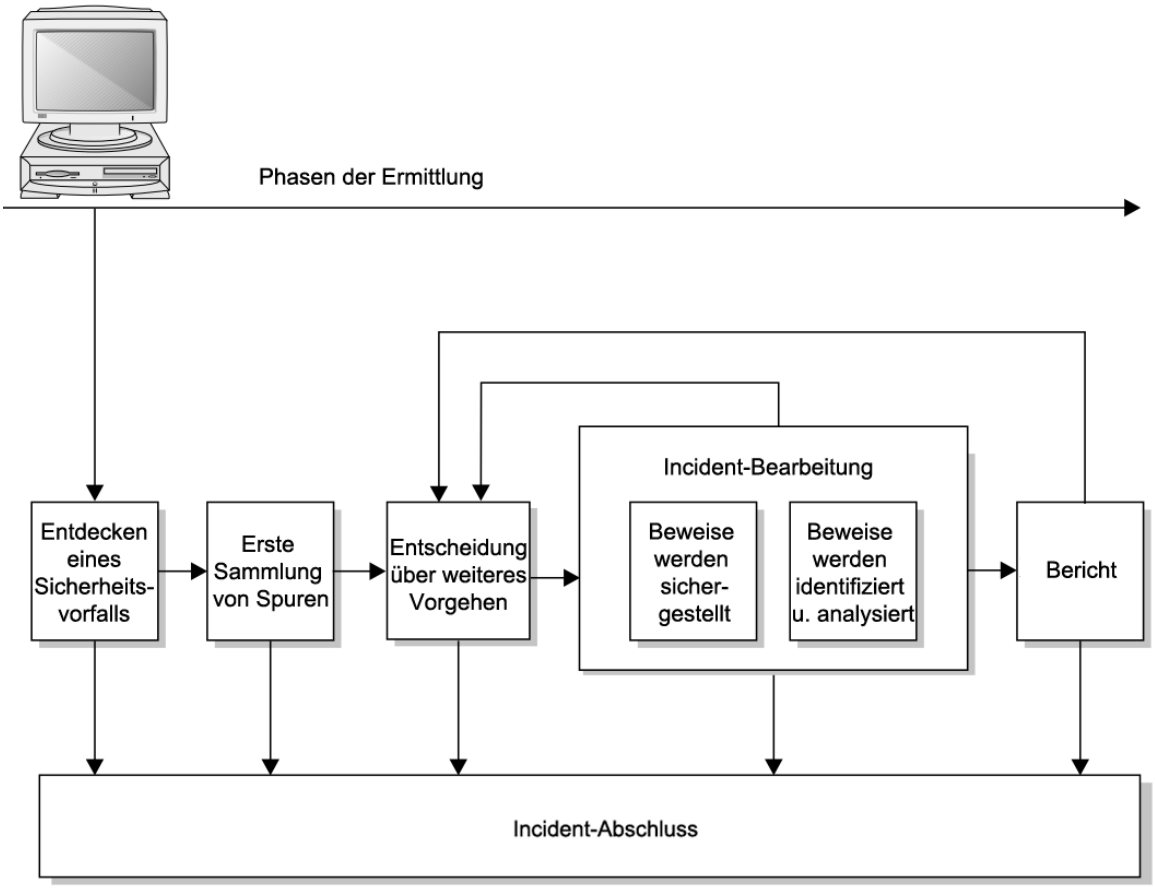

Abbildung 4.1.: Schematische Darstellung des Ermittlungsprozesses

Die eigentliche forensische Untersuchung beginnt mit der umfangreichen Datensammlung, in welcher alle Indizien und Beweise identifiziert und gesichert werden. Hierbei wird darauf geachtet, dass der Untersuchungsbereich sorgfältig abgesichert und dass die Beweissicherung forensisch korrekt durchgeführt wird - Daten dürfen nämlich nicht durch die Sicherungsmaßnahmen verfälscht werden [vgl. Abschnitt 4.2]. Hierfür wird von dem Vier-Augen-Prinzip und dem MD5-Hash-Verfahren Gebrauch gemacht. Durch die richtige Reihenfolge bei der Beweissicherung wird der Grundstein für eine erfolgreiche Aufklärung gelegt, Daten werden in der Reihenfolge ihrer Flüchtigkeit gesammelt. Meistens ist eine Analyse eines Sicherheitsvorfalls leichter möglich, wenn von kompromitierten IT-Systemen ein forensisches Duplikat der Datenträger angefertigt und damit die Gefahr eingedämmt wird, dass die Beweise unabsichtlich geändert werden. Diese Kopien ermöglichen es, ohne Zeitdruck an der Beweissammlung zu arbeiten, während auf den betroffenen Systemen Original-Konfigurationen aus Backups wiedereingespielt werden. Prinzipiell werden bei der Durchführung der Datensammlung alle Tätigkeiten in einem Protokoll sorgfältig festgehalten.

In der Datenanalyse-Phase (vgl. Abbildung 4.1) werden die Beweisspuren sorgfältig untersucht und analysiert. Da alle betroffenen IT-Systeme insgesamt als unsicher oder manipuliert betrachtet werden müssn, legt dieser Schritt den Fokus darauf, aus den zu diesem Zeitpunkt gesammelten Daten forensisch wertvolle Hinweise zu extrahieren und die Zusammenhänge zwischen einzelnen Spuren herzustellen. Hierzu gehören insbesondere die Untersuchung der Betriebssystem-, Konfigurations- und Benutzerdateien auf Manipulationen sowie die Betrachtung aller relevanten Log-Einträge. Die detaillierte Vorgehensweisen werden in den nachfolgenden Kapiteln erläutert.

Schließlich umschreibt „Dokumentation“ ein Konzept zur ausführlichen Protokollierung alle durchgeführten Aktionen. Entscheidend ist hierbei die Unterscheidung der Maßnahmen zur Dokumentation in „prozessbegleitende Dokumentation“ und „abschließende Dokumentation“.

„Prozessbegleitende Dokumentation“ umfasst prinzipiell Protokollierung der eingeleiteten Untersuchungsmaßnahmen sowie der gewonnenen Daten zur Übereinstimmung mit den Prinzipien der Forensik [4.2] sowie zur Einhaltung der Anforderungen an den Ermittlungsprozess. Typische Parameter sind beispielsweise

- Name und Versionsnummer des verwendeten Programms
- Kommandozeilenparameter des Aufrufs
- Maßnahmen zur forensischen Absicherung

Die „abschließende Dokumentation“ beinhaltet die Beschreibung des gesamten Ermittlungsprozesses und stellt durchgängige Zusammenhänge aus den untersuchten Daten her. Ziel der „abschließenden Dokumentation“ ist es sicherzustellen, dass wichtige Untersuchungsergebnisse auch für Dritte nachvollziehbar bleiben und gleichzeitig den Entscheidungsträgern die Möglichkeit zu eröffnen die Ermittlungsergebnisse den Strafverfolgungsbehörden zu präsentieren.

4.4. Einordnung der IT-Forensik in den SIR-Prozess

Die Konzeption des IT-Forensik-Leitfadens ist LRZ-spezifisch und bezieht die Verfügbarkeitsanforderungen mit ein. Eine Herausforderung ist die enge Verflechtung mit dem Security-Incident-Response-Prozess - bei einem Sicherheitsvorfall stehen nämlich unterschiedliche Prioritäten im Vordergrund. Während die IT-Administratoren in Anlehnung an gut geplanten SIR-Prozess versuchen, die Reaktionszeit auf ein Vorfall zu minimieren und den entstandenen Schaden einzudämmen, besteht für die IT-Forensiker die Herausforderung darin, die Lage gewissenhaft zu analysieren, Beweise und Indizien zu identifizieren sowie zu sichern, um Einsichten ín die Vorgehensweise des Angreifers zu gewinnen. Die Bewahrung des Status Quo steht daher an erster Stelle.

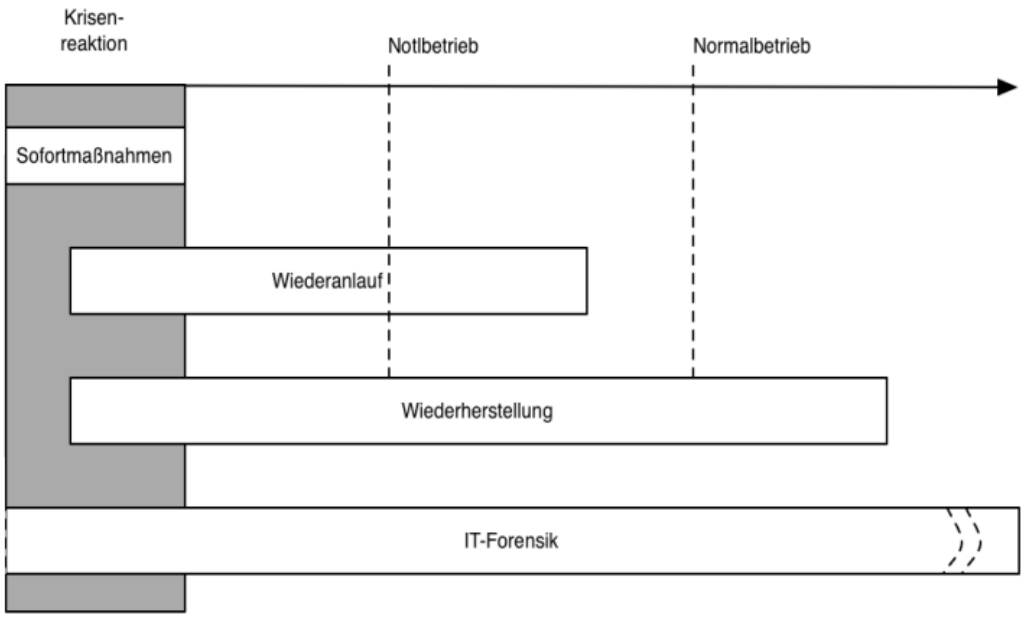

Abbildung 4.2.: Zeitliche Einordnung der IT-Forensik in den SIR-Prozess

Quelle: [BSI11b, S. 34]

Erfahrungen aus der Praxis zeigen: Computerforensik agiert als Bindeglied zwischen der Incident Response auf der einen Seite und der Einleitung der juristischen Schritte auf der anderen Seite. Oft ist die Durchführung einer Untersuchung unter der Hinzunahme der IT-Forensik mit der Reaktion auf einen Notfall im Rahmen des SIR-Prozesses gleichzusetzen (siehe Abb. 4.2). Bild 4.5 zeigt die schematische Darstellung der Security-Incident-Response-Prozesses am LRZ unter der Berücksichtigung der Computer-Forensik.

Der Nutzen der Computer-Forensik wird bei einer Betrachtung der juristischen Zusammenhänge deutlich. Erst nach und nach können bei einer Störung Schäden sowie Folgeschäden quantifiziert werden, so dass eine Entscheidung über eine straf- bzw. zivilrechtliche Verfolgung des Verursachers erst mitten in der Untersuchung getroffen werden kann. Wurden bis zu diesem Zeitpunkt bereits vorgenommene Schritte nicht entsprechend den forensischen Richtlinien [4.2] durchgeführt und dokumentiert, ist der Fall verloren. Eine methodische Vorgehensweise unter Einbeziehung der Möglichkeiten der Computer-Forensik ist daher ausschlaggebend.

Die Einordnung der Computer-Forensik erfolgt in Anlehnung an [Met11] - Forensikmaßnahmen müssen nahtlos in die bestehenden Prozessabläufe eingeordnet werden und sollten gleichzeitig weder Aufwand noch Kosten in die Höhe treiben. Wie immer gilt: je komplexer und zahlreicher umzusetzende Maßnahmen, desto höher Aufwand und Kosten. Dabei ist der Weg vom Feststellen eines Security-Incidents bis zum Vorlegen von Beweisen anhand des folgenden Ablaufs festgelegt:

Incident-Erfassung und -Aufnahme

- Ein Sicherheitsvorfall bzw. eine eine sicherheits-relevante Auffälligkeit wird erkannt und an das LRZ-CSIRT gemeldet.
- Allgemeine Informationen zum Vorfall werden aufgenommen und in iET ITSM in einem Security-Incident-Ticket erfasst.

Initiale Klassifikation durch CSIRT-Hotliner und SIC

- Die Informationen zu einem Sicherheitsvorfall werden durch den CSIRT-Hotliner und SIC nach verschiedenen Gesichtspunkten geprüft (aktuelle Auflistung der Kriterien kann [Met11] entnommen werden):
  - Liegt ein Security-Incident (SI) oder ein Fehlalarm vor? Falls SIC und CSIRT-Hotliner feststellen, dass es sich um einen Fehlalarm oder eine Betriebsstörung handelt, wird der Vorfallmelder entsprechend informiert und das Ticket geschlossen oder an die zuständige Abteilung zur Bearbeitung weitergeleitet.
  - Handelt sich um ein SI, muss die Anzahl betroffener Systeme bestimmt werden.
  - Zu ermitteln sind Antworten auf die Fragen: Welche Abhängigkeiten (Daten/Dienste) sind vorhanden? Sind kritische Dienste betroffen? Ist eine Ausweitung zu einem Major Incident zu erwarten?
  - Eine der wesentlichen Erkenntnisse bei der Eindämmung eines Sicherheitsvorfalls ist die Frage, ob es sich um ein Standard-Security-Incidents (SSI) handelt.
  - Zu dokumentieren ist welche System- und Netzadministratoren für das gemeldete IT-System verantwortlich sind.
  - Für die weitere Bearbeitung des Vorfallst ist die Bestimmung des Zeitpunkts des ersten Auftretens unter Beachtung von unterschiedlichen Zeitzonen wichtig.
- Eine der grundlegenden Entscheidungen durch SIC betrifft die Bestimmung der Auswirkung und Dringlichkeit des Incidents anhand folgender Kriterien:
  - Wo befinden sich betroffene/gefährdete Zielsystem(e)?
  - Welche Dienste/Daten sind mittelbar oder unmittelbar betroffen?
  - Wieviele Systeme sind betroffen?
  - Was ist die Quelle des Angriff?

- Die Aufschlüsselung der einzelnen Kriterien sind der Abbildung 4.3 zu entnehmen, Zahlen in Klammern gegeben die konkreten Werte für das jeweilige Kriterium an. Eine einfache Gesamtsumme, die die Werte in einer einzelnen Spalte addiert, ergibt die Auswirkung des Vorfalls. Daraus ergibt sich automatisch, anhand der Abbildung 4.4 die Priorität des Vorfalls.

Auswirkung/ Kriterium	Niedrig	Mittel	Hoch
Standort (Zielsystem)	Extern (1)	MWN, Grid-Systeme (2)	LRZ-intern (3)
Dienste	Kritische Dienste sind nicht betroffen (1)	Dienste für MWN und Grid-Umgebung (2)	Wichtige LRZ-Dienste (Mail, DNS, DHCP, ...) (3)
Daten	Kritische Daten sind nicht betroffen (1)	Daten aus MWN bzw. Grid betroffen (2)	Kritische LRZ-Daten sind betroffen (3)
Anzahl betroffener Systeme	1 (1)	2-3 (2)	> 3 (3)
Standort (Quellsystem)	Extern (1)	MWN, Grid-Systeme (2)	LRZ-intern (3)

Abbildung 4.3.: Bestimmung der Auswirkung eines Security-Incidents anhand vordefinierter Kriterien.

Quelle: [Met11]

Auswirkung	Gering	Mittel	Groß	Sehr groß
Berechneter Wert	5,6	7,8,9	10,11,12	13,14,15

Abbildung 4.4.: Bestimmung der Priorität eines Security-Incidents.

Quelle: [Met11]

- Ferner hat SIC zu klären, ob eine forensische Untersuchung des Vorfalls notwendig ist. Dies ist immer eine Abwägungsfrage, welche u.a. im zeitlichen Aufwand begründet ist. Es müssen für jeden Einzelfall folgende Fragen beantwortet werden:
  - Handelt es sich um ein „produktives“ System?
  - Ist das betroffene System **kritisch** für das LRZ?
  - Entsteht durch den Systemeinbruch und damit verbundenen Produktivitätsausfall ein (hoher) finanzieller Schaden?
  - Kann Methode oder Schwachstelle, die zum Systemeinbruch geführt hat, ohne tiefergehenden Nachforschungen identifiziert werden?
  - Ist forensische Sicherung von Daten zur potentiellen straf- oder zivilrechtlichen Verfolgung notwendig?
- Die Beantwortung dieser Fragen ist (vor allem zu diesem Zeitpunkt) oft schwierig und muss häufig auf Basis von Annahmen, bisherigen Erfahrungen und Schätzungen erfolgen. Grundsätzlich gilt: kann eine der Fragen „bejaht“ werden, sollten forensischen Maßnahmen ergriffen werden. Eine Tatsache sollte nicht aus den Augen verloren werden: selbst wenn eine gerichtliche Auseinandersetzung nicht im Vordergrund steht, kann eine geordnete Vorgehensweise unter Einbeziehung der Computer-Forensik schnellere Ergebnisse liefern.

Incident-Bearbeitung, Analyse und Diagnose betroffener Systeme

- **Anmerkung:** Bei einem Standard-Security-Incident oder einem Vorfall mit niedriger Priorität kann eine abweichende, nichtstandardkonforme Vorgehensweise gewählt werden.

- Im Rahmen von SIR sind grundsätzlich immer die Fragen nach Identität des Angreifers, Ausmaß der Bedrohung/der Schäden und den Schwachstellen, die zur Kompromittierung des Systems geführt hatten, sowie ob weitere Systeme in der Umgebung betroffen sind zu klären.

- Welche Maßnahmen eingeleitet werden können, hängt wesentlich davon ab, welche Ergebnisse erste Analyse der betroffenen Systeme durch zuständige Administratoren liefert - eine umfassende Kenntnis über kompromitierte IT-Infrastruktur bildet den Ausgangspunkt für notwendige Maßnahmen. Die Tiefe der Analyse hängt von der Priorisierung des Vorfalls ab. Den Administratoren stehen dabei vorgefertigte Toolkits zur Verfügung, die sie (bei Bedarf) bei der Sammlung von Spuren unterstützen und diese gleichzeitig mit der größtmöglichen Beweishärte sicherstellen. Eine genaue Vorgehensweise sowie Rollenaufteilung kann dem Kapitel 5 entnommen werden - es wird detailliert auf Systemzustand (aktiv/ausgeschaltet) sowie Systemart (physisch/virtuell) eingegangen.

- In regelmäßigen Intervallen wird detaillierte Rückmeldung an das LRZ-CSIRT erwartet. Alternativ können die Analyseergebnisse von einem CSIRT-Hotliner angefordert werden. Eine ausführliche Dokumentation des Vorfalls ist unerlässlich.

- Ein wichtiger Punkt des SIR-Prozesses ist die Auswertung der gesammelten Daten durch das LRZ-CSIRT bzw. einen erweiterten Personenkreis. Eine detaillierte - auf eine konkrete Situation abgestimmte - Auswertung der Ergebnisse ist unerlässlich - erst wenn aktuelles Bedrohungsszenario analysiert ist, kann der SIC über weiteres Vorgehens entscheiden. Dabei spielen eine Vielzahl von weiteren Fragestellungen eine wichtige Rolle, z. B. Verfügbarkeits-Überlegungen, einzusetzende Technologien oder der Umgang mit Beweismitteln. Weitere Maßnahmen, die durch Mitglieder des LRZ-CSIRT bzw. zuständige Administratoren umzusetzen sind, könnten beispielsweise so aussehen und werden zum Teil im Kapitel 6 ausführlich beleuchtet:

  - Analyse des Arbeitsspeicherdumps,
  - Erstellung eines forensischen Duplikats und anschließende Post-Mortem-Analyse ,
  - Re-Priorisierung des Sicherheitsvorfalls,
  - Neuinstallation betroffener Systeme,
  - ...

- Bei einer forensischen Untersuchung wird dem Schutz der Intergrität der Beweismittel hohe Bedeutung beigemessen [5.9.2]. Der zuständige CSIRT-Forensiker agiert als Sachwalter, nimmt Beweisspuren entgegen und verwahrt sie sicher auf. Vom Vier-Augen-Prinzip soll ausgiebig Gebrauch gemacht werden.

Lösung und Wiederherstellung des Regelbetriebes

- In der letzten Phase geht es darum, schnellstmöglich zum betrieblichen Alltag überzugehen. Je nach Situation wird eine oder mehrere Maßnahmen (z.B. Entfernung der Schadsoftware oder Anpassung der Systemkonfiguration) durch das CSIRT angestoßen, wobei möglicherweise Maßnahmen während der Umsetzung angepasst werden müssen.

- Anschließend erfolgt im laufenden Betrieb eine Überwachung der betroffenen Systeme sowie das Überprüfen der Maßnahmen und bei erneutem Auftreten von sicherheitsrelevanten Vorfällen das angemessene Reagieren. Die Überwachungsdauer wird vom SIC festgelegt und beträgt im Regelfall 5 Werktage.

- Falls eine forensische Untersuchung durchgeführt wurde, wird vom zuständigen CSIRT-Hotliner in Zusammenarbeit mit den System- bzw. Netzverantwortlichen ein Abschlussbericht erstellt.

Obwohl der Nutzen, der sich aus einem strukturierten IT-Forensik-Leitfaden ergibt, eigentlich offensichtlich ist, bleiben in der Praxis oft Unklarheiten bestehen. Diesem Mangel kann in der Evaluierungsphase des Leitfadens entgegengewirkt werden. In dieser Phase haben erprobte Methoden und Vorgehensweisen weiterhin Bestand - Paradigmen, die sich über Jahre bewährt haben, sollen nicht verworfen werden.

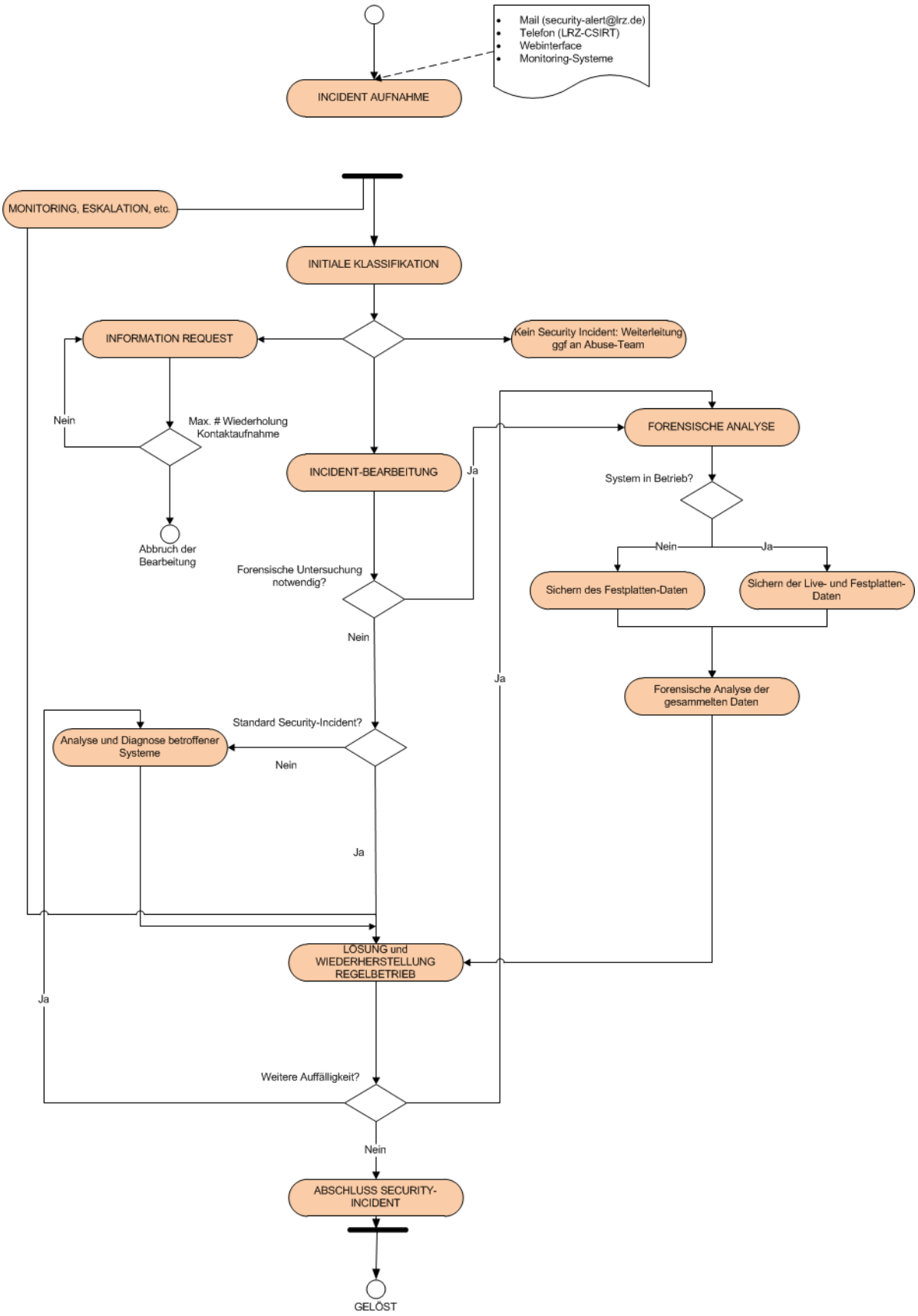

Abbildung 4.5.: LRZ-SIR-Prozess unter Hinzunahme der IT-Forensik

5. Forensische Datenerfassung

Nach den vorangegangenen Kapiteln, die einen groben Überblick über die verschiedenen Zusammenhänge gaben, bildet eine klar definierte, nach den LRZ-Anforderungen ausgerichtete Verfahrensweise für die forensische Datenerfassung den Mittelpunkt dieses Kapitels. Eine grundlegende Vorbereitung ist eine der Voraussetzungen für eine erfolgreiche Ermittlung - die entsprechenden Teilaspekte werden daher ausführlich beleuchtet. Auf technischer Seite wird, sofern notwendig, zwischen Windows Server 2008R2 und SuSE Linux Enterprise Server unterschieden. Genauso wichtig ist Kenntnis über Sicherung der flüchtigen Daten vom untersuchten IT-System.

Ein nicht minder weites Feld ist das Thema „Live Response“. Die Teilaspekte der „Live Response“ umfassen u.a. die Erfassung von Beweisspuren auf einem noch „lebenden“ System. Im Rahmen dieses Unterkapitels werden wichtige Fragestellungen erklärt.

Das Kapitel beleuchtet darüber hinaus Datenerfassung im Netzwerk sowie das Anfertigen eines forensischen Duplikats. Der letzte Abschnitt liefert Hinweise für den korrekten Umgang mit Beweismitteln.

5.1. Datenerfassung vom noch „lebenden“ System

Das infizierte System herunterzufahren und ein Duplikat der angeschlossenen Laufwerke anzufertigen, war lange Zeit der zentrale Bestandteil der Computer-Forensik - doch mittlerweile kann das geordnete Herunterfahren von Systemen einen finanziellen Schaden im drei- oder vierstelligen Betrag pro Minute bedeuten. Die Cyber-Angriffe gewinnen an Raffinesse, die international organisierten Banden bringen erstaunliches Know-how mit und tarnen ihre Angriffe und Schadcodes. Seit einiger Zeit sind Evasion-Techniken bekannt, bei denen der Programmcode nie auf die Festplatte geschrieben wird, sondern direkt aus dem Internet nachgeladen und im Arbeitsspeicher ausgeführt wird [Car09]. Diese Methoden hebeln die traditionelle Vorgehensweise der forensischen Duplikation teilweise aus: Wird das betroffene System heruntergefahren, würden wichtige Spuren für immer verloren gehen.

Eine besondere Beachtung gebührt bei der Behandlung von Sicherheitsvorfällen darüberhinaus der Trennung eines noch „lebenden“ IT-Systems von der Spannungsversorgung und/oder vom Netzwerk. „RFC 3227 - Guidelines for Evidence Collection and Archiving: Things to avoid“ [RFCa] legt den Ermittlern nahe, das angegriffene System so lange laufen zu lassen, bis alle wertvollen Informationen gespeichert sind. Die endgültige Entscheidung hängt von vielen Faktoren ab und muss im Kontext des konkreten Falls getroffen werden. Soll das System vom Stromnetz genommen werden, ist es ratsam zum sog. harten Shutdown zu greifen - das System soll durch einfaches Steckerziehen heruntergefahren werden. Das Risiko für das eigentliche System ist kalkulierbar und kann als gering eingestuft werden. Beim ordnungsgemäßen Shutdown werden die Zeitstempel des letzten Zugriffs vieler Dateien verändert, infizierte Systeme sind häufig mit Schadsoftware belastet, die beim Herunterfahren eines Computersystems aktiv wird.

Im September 2011 hat Seagate die weltweit erste Festplatte mit einer Kapazität von 4 TByte vorgestellt [CB]. Am LRZ werden als zentrale Speichermedien redundant ausgelegte, hochperformante RAID-5 und RAID-10 Festplattenverbunde in Kombination mit zentralen NAS-Speicherquellen eingesetzt. Ermittler würden mehrere Stunden (bzw. bei bestimmten Konfigurationen Tage) brauchen, um ein forensisches Duplikat anzufertigen. Technisch kann die schiere Datenmenge die forensische Duplikation schwierig machen - bei der Festlegung der Vorgehensweise sollten die Speicherplatzanforderungen ebenfalls mitberücksichtigt werden, um Engpässe vorzubeugen.

Zum besseren Verständnis der allgemeinen Zusammenhänge empfiehlt sich Betrachtung zweier grundlegenden Sachverhalten: der Locard’schen Regel und der Halbwertzeit der Daten. Diese Aspekte stehen in direktem

Verhältnis zu den Tätigkeiten bei den Ermittlungen.

Locard'sche Regel Einen wichtiger Teilaspekt der Forensik bildet die Locard'sche Regel, benannt nach dem berühmten Kanadier Edmond Locard. Sie besagt, dass keine Aktion von einem Täter vollzogen werden kann, ohne eine Vielzahl von Spuren zu hinterlassen [Wike].

Das gleiche Prinzip gilt auch in der digitalen Welt. Hier gilt es zu berücksichtigen, dass zur Laufzeit diverse Änderungen im Hostsystem auftreten, bedingt durch Prozessausführung oder Dateioperationen, beispielsweise im Arbeitsspeicher oder im persistenten Speicher. Dies gilt auch für Zeiträume der Inaktivität, solange sich das System nicht im Ruhemodus befindet. Diese Zustandsänderungen lassen sich mit der Locard'schen Regel erklären und werden unter dem Begriff „Beweisdynamik" zusammengefasst - am ehesten lässt sich dieses Phänomen mit Regen vergleichen, der am Tatort mögliche Beweise wegwischt bevor sie erfasst wurden.

Auch während einer forensischen Untersuchung am verdächtigen System treten unweigerlich Veränderungen des Systemzustands auf. Man sollte sich bewusst sein, dass fast jedes Programm, das man zu Ermittlungszwecken auf dem betroffenen System startet, Spuren hinterlässt. Das Ausführen von Tools zur Datenaquisition birgt die Gefahr, dass im Arbeitsspeicher bzw. im Cache zwischengespeicherten Daten bzw. Datenfragmente durch neuere Aktivitäten überschrieben werden und damit relevante Informationen zerstört werden bzw. verschleiert werden können. Diese zusätzlichen Spuren stören dann die Auswertungslogik. Die Konsolenapplikation zur Erstellung der Speicherabbilder WinEn [5.5.2.2], Teil der Forensiksuite EnCase, legt beispielsweise im Startverzeichnis eine Treiberdatei mit dem Namen *winen_sys* ab. Zusätzlich wird in der Windows-Registrierungsdatenbank ein neuer Systemdienst mit der Bezeichnung *winen_* angelegt. Der Dienst wird bei jedem Reboot automatisch gestartet. Das bedeutet, dass ein Eintrag im Windows Eventlog hinterlassen wird [Mul]. Die Ermittler müssen nicht nur berücksichtigen, dass Zustandsänderungen bei Zugriffen unvermeidlich sind, sondern sie müssen gleichzeitig in der Lage sein, die Auswirkungen ihrer Aktionen auf das IT-System vorauszusagen, sie zu dokumentieren und angemessen erklären zu können. Vor diesem Hintergrund ist es nicht überraschend, dass eine zeitnahe und behutsame Speicherung des Hauptspeicherinhalts im Vordergrund steht.

Halbwertzeit der Daten Ein weiteres Thema der Datenaquisition vom „lebenden", verdächtigen System ist die Halbwertszeit der Daten. Grundsätzlich lassen sich einige empfindliche Datenarten identifizieren, deren Haltbarkeit sehr unterschiedlich sein kann.

„RFC 3227" [RFCa] versucht diese unterschiedliche Datentypen nach ihrer Halbwertszeit zu sortieren, um die Reihenfolge der Datensammlung zu bestimmen. Von besonderem Interesse ist der Abschnitt 2.1 „Order of Volatility". Idealerweise sollten die Daten in der Reihenfolge ihrer Halbwertszeit gesichert werden. Dafür ist es erstrebenswert, die flüchtigen Daten (Inhalt von Cache und Hauptspeicher, Status der Netzverbindungen, laufende Prozesse, angemeldete Benutzer, usw.) schnell und strukturiert zu sichern. Von sämtlichen (weniger) volatiler Daten (Konfigurationsdateien des physischen/virtuellen IT-Systems, Logeinträge) kann zu einem späteren Zeitpunkt behutsam eine Kopie angefertigt werden.

Obwohl bereits im Jahr 2002 veröffentlicht, bleiben diese Richtlinien für die Spurensicherung weiterhin aktuell.

Datentypen Unabhängig davon lassen sich einige Informationsquellen bei einem kürzlich angegriffenen System identifizieren, die einerseits für die Ermittlung von Interesse sind und andererseits beim Herunterfahren von IT-Systemen verloren gehen würden. Konkret handelt es sich hierbei um:

- Aktuelle Uhrzeit (mit Abweichung von einer vertrauenswürdigen Referenzzeit[1]),
- Inhalt des Hauptspeichers,
- Liste der laufenden Prozesse,
- Liste der angemeldeten User,

[1] Dadurch ist es möglich bei der späteren Auswertung eines forensischen Duplikats eigene Änderungen am System von denen des Angreifers zu unterscheiden. Zu beachten ist, dass bei jedem Zugriff auf eine lokale Datei sich deren MAC-Time ändert.

- Liste der geöffneten Sockets,
- Liste der Anwendungen, die auf geöffneten Sockets lauschen,
- Liste der Systeme, die gerade eine Netzverbindung haben oder vor Kurzem eine hatten,
- Cache-Inhalt.

5.2. Forensik-Workstation

Das Testsystem besteht aus Hard- und Software zur Durchführung von forensischen Analysen. Bei dem System handelt es sich um eine Windows Server 2008R2 virtuelle Maschine, die auf VMware ESXi aufsetzt. Für die Rechenleistung sorgt eine virtuelle CPU sowie 3072 MByte Hauptspeicher. Um das System in ein Netzwerk einzubinden, verfügt dieses über einen virtuellen Gigabit-Controller. Das VM ist ferner mit zwei virtuellen Festplatten (je 40 GB) ausgestattet, die von einem SCSI-Controller verwaltet werden - zusätzlich lässt sich Netzspeicher einbinden. Darüber hinaus steht ein DVD-ROM-Laufwerk zur Verfügung. Die Laufwerkskonfiguration wird mit einem Floppy-Gerät zur Einbindung von virtuellen Disketten abgerundet.

Um Ressourcenknappheit zu vermeiden, lassen sich die Konfigurationswerte jederzeit anpassen - Anzahl der virtuellen CPUs, Größe des Haupt- sowie des Festplattenspeichers und die Konfiguration des Netzes können von den zuständigen VMware-Administratoren nach einer Anfrage modifiziert werden. Von der Flexibilität bei der Zusammenstellung des Systems abgesehen, funktioniert die Forensik-Workstation wie jedes standardmäßige System.

Vorteile der Verwendung einer VM für forensische Untersuchungen entfalten sich typischerweise in zwei Phasen:

- Sobald das System eingerichtet ist, kann es beliebig oft repliziert werden - schnelle und einfache Inbetriebnahme beziehungsweise Konfiguration sind die Folge.
- Der eigentliche Nutzen zeigt sich mit immer weiter steigenden Verbreitung von virtuellen Maschinen am Leibniz-Rechenzentrum - bei Bedarf lassen sich beispielsweise virtuelle Festplatten direkt einbinden. Der Kreativität sind dabei kaum Grenzen gesetzt - ohne zu weit vorauszugreifen, ist es, abhängig vom Standort des virtuellen Systems, nicht ohne Anpassung der Konfiguration der Netzinfrastruktur möglich, forensische Daten an das Analysesystem zu übertragen. In diesem Fall kann, z.B. zur Anfertigung eines forensischen Duplikats 5.8, ein zusätzlicher Datenträger an die kompromittierte virtuelle Maschine angeschlossen werden, welches anschließend an das Analysesystem angeschlossen werden kann.

5.3. Live-Response-Methodik

In diesem Unterkapitel sollen nun grundlegende Eckpunkte der Live-Response-Methodik diskutiert werden, die sich für eine forensische Untersuchung eines laufenden Vorfalls ergeben. Die Art des Zugriffs auf das betroffene System spielt bei der Datensammlung eine wichtige Rolle. Üblicherweise werden Bestandteile der eigens zusammengestellten Toolsammlung von einem schreibgeschützten Medium (ISO-Datei/USB-Stick mit Schreibschutz/CD) nacheinander gestartet, die Ergebnisse werden über Netz auf ein File-Share geschrieben. Wichtiger Vorteil: es wird nur mit eigenen sicheren und aus vertrauenswürdiger Quelle stammenden Dateien gearbeitet, es müssen keine clientseitige Vorbereitungen getroffen werden und es müssen keine externe Datenträger an das betroffene System angeschlossen werden.

Ein weiterer wesentlicher Vorteil: Durch die gewählte Vorgehensweise werden Interaktionen mit dem betroffenen System und damit auch Veränderung der volatilen Daten auf ein Minimum reduziert - damit wird der Locard'schen Regel [5.1] angemessen Rechnung getragen - immerhin hinterlässt jede Aktion Spuren.

Bei der eingesetzten Software zur Sicherstellung der volatilen Daten ist zu unterscheiden zwischen Windows- und Linux-Toolkit. In ihrem Grundprinzip sind sie ähnlich aufgebaut: sie setzen auf eine eigene vertrauenswürdige, externe Shell, das Aufrufen von Tools ist durch Stapelverarbeitung automatisiert. Gegenüber

der händischen Datensammlung bieten die Toolkits reduzierte Fehlerwahrscheinlichkeit, viele CL-Befehle werden selbstständig aufgerufen und abgearbeitet. LRZ-Administratoren sind ferner nicht mehr auf Systembefehle (die trojanisiert sein könnten) angewiesen - sie können tiefergehende Untersuchungen mit aus vertrauenswürdigen Quellen stammenden Tools durchführen. Ein weiterer wesentlicher Vorteil: eigene Software-Lösung ermöglicht eine flexible und an die Bedürfnisse des Leibniz-Rechenzentrums angepasste Implementierung und kann je nach Bedarf erweitert bzw. modifiziert werden. Bedingt durch die sehr komplexe IT-Infrastruktur und sehr spezifische Implementierungsanforderungen ist Einsatz eines „Rundum-sorglos-Pakets“ nur schwer vorstellbar.

Unterschiede zwischen den Linux- und Windows-Toolkits sind in erster Linie implementierungsbedingt: Bereits bei oberflächlicher Betrachtung wird man feststellen, dass die Linux-Kommandozeile wesentlich mächtiger ist und weitaus mehr Funktionen bietet als die Windows-Eingabeaufforderung. Dazu gehört die Verarbeitung von Bedingungen, die Unterstützung von Funktionen und die Fähigkeit, Schleifen zu bilden. Dadurch ist die Implementierung des Linux-Toolkits wesentlich eleganter und effizienter. Weiterführende Informationen können den jeweiligen Source-Codes entnommen werden - beide sind ausführlich dokumentiert.

5.4. Prozessbeschreibung

In diesem Abschnitt werden das allgemeine Vorgehen und die Verantwortlichkeiten geregelt. Es wird festgehalten, welche flüchtige Daten und vor allem mit welchen Tools diese gesichert werden sollen und wie sie zu archivieren sind. Live Response ist ein Prozess, der nur durch einen klar definierten Personenkreis durchzuführen ist. Das gilt sowohl für die Datenerfassung als auch für die Sicherung von Beweismitteln und Durchführung der späteren Datenanalyse.

Eine Live Response, die regelmäßige manuelle Eingriffe erfordert, ist auf lange Sicht sehr umständlich. Entsprechende Prozessschritte werden deshalb so weit wie möglich automatisiert. Dazu gehört in erster Linie das skriptgesteuerte Sammeln entsprechender Spuren.

- Die Entscheidung, ob Live-Response durchgeführt werden soll, wird durch den SIC in Abhängigkeit von der Priorität und Klassifikation des Vorfalls getroffen. Die Kriterien, die es zu berücksichtigen gilt, wurden bereits im Abschnitt 4.4 vorgestellt.
- Die Art des betroffenen Systems spielt beim Sammeln von forensischen Daten eine wichtige Rolle. Bei der Wahl einer geeigneten Vorgehensweise sind Eigenheiten der physischen sowie virtuellen Systemen bzw. des verwendeten Betriebssystems (Windows Server 2008R2/SLES 11) zu berücksichtigen.
- Der aktuelle Systemzustand spielt bei der Datensammlung eine wichtige Rolle. Es sind folgende Kriterien zu berücksichtigen:
  - System läuft noch/System ist ausgeschaltet,
  - Netzanbindung ist aktiv/Netzanbindung wurde unterbrochen.
- Sicherung der gesammelten Daten erfolgt auf dem zentralen „Forensik-Server“.

Tritt ein SI ein, so gilt es, so schnell wie möglich genügend Informationen für eine forensische Analyse zu sammeln. Dies erfolgt in der Regel durch zuständige Systemadministratoren. Abhängig von der Systemart bzw. aktuellem Systemzustand variiert dabei die Vorgehensweise:

5.4.1. Physisches System

Bei einem physischen System entspricht die Vorgehensweise zur sofortigen Sicherung aller wichtigen Spuren in etwa der klassischen Computer-Forensik, wie in der Abbildung 5.1 zu sehen ist. Alle Tätigkeiten sollten mit genauer Uhrzeit protokolliert werden.

System läuft noch

- Beim Vorliegen eines Security-Incidents dürfen systemeigenen Befehle nicht verwendet werden. Für Untersuchungen von physischen Maschinen wird eine Toolsammlung vom CSIRT-Team gepflegt. Folgende Möglichkeiten zur Einbindung stehen dabei zur Verfügung:
  - Laden von einer CD,
  - Laden von einem schreibgeschützten USB-Stick,
  - Laden von einem schreibgeschützten Network-Share.
- Nach dem Einbinden wird eine vertrauenswürdige Shell aufgerufen und das Skript zur automatisierten Datensammlung gestartet. Daten werden dabei in der Reihenfolge ihrer Flüchtigkeit gesammelt:
  - aktuelle Systemzeit und deren Differenz zu einer vertrauenswürdigen Referenzzeit erfassen
  - Bei $\geq$ „Mittel“ priorisierten Sicherheitsvorfällen - Sicherung des Hauptspeicherinhalts,
  - Informationen über eingeloggte Benutzer,
  - Informationen über Netzverbindungen,
  - Informationen über laufende Prozesse,
  - aktuelles Portmapping,
  - Informationen über aktuellen Netzstatus,
  - Sicherung verschiedenster Systeminformationen (Eventlogs, Konfigurationsdaten, etc),
  - Sicherung des Prozessspeichers (nur Linux).
- Da Ergebnisse der Untersuchung auf einem zentralen „Forensik-Server“ gesichert werden, muss der zuständige CSIRT-„Sachwalter“ (siehe Abschnitt 4.4) Daten entgegennehmen und Prüfsummen zum Schutz der Integrität der Beweise berechnen.
- Parallel dazu kann das LRZ-CSIRT manuelle Analyse des Kommunikationsverhaltens des betroffenen Systems durchführen. Im Wesentlichen spielen folgende Faktoren eine Rolle:
  - Auswertung der Netflow-Daten kann zur Bestimmung der Tragweite des SI herangezogen werden,
  - Daten über Kommunikationsverhalten können wertvolle Hinweise für weitere Analyse liefern (ungewöhnliches Kommunikationsverhalten, offene Ports, usw).
- Alle wichtigen Spuren werden sofort analysiert. im Rahmen einer genauen Auswertung durch CSIRT sowie zuständige Administratoren muss die weitere Vorgehensweise durch den SIC festgelegt werden. Je nachdem wie ergiebig die Analyseergebnisse sind, bieten sich im Wesentlichen folgende Maßnahmen an (die durchaus auch kombiniert zum Einsatz kommen können):
  - Das System kann heruntergefahren werden - es soll durch „einfaches Steckerziehen“ abgeschaltet werden. Ein ordnungsgemäßes Herunterfahren ändert MAC-Zeitstempel unzähliger Dateien, außerdem können einige Dateien gelöscht oder neu anlegen werden - das würde nicht reallozierten Platz gelöschter Dateien überschreiben und Wiederherstellung unmöglich machen.
  - Datenträger enthalten eine Vielzahl kritischer Daten - im Rahmen der forensischen Untersuchung kann entweder eine selektive Sicherung für den Fall relevanter Dateien oder vollständige forensische Duplikation der angeschlossenen Datenträger durchgeführt werden. Bei einem *forensischen Duplikat* [5.8] wird der gesamte Datenträger bitweise auf Datensicherungsmedien übertragen. Der Vorteil dieses Verfahrens liegt in der Vollständigkeit der gesicherten Daten, selektive Extraktion der Daten und der Möglichkeit der tiefgründigen Datenträgeranalyse. Nachteilig ist der bei großen Datenmengen hohe Zeitbedarf und der Speicherplatzverbrauch. Typischerweise wird forensische Duplikation nur bei $\geq$ „Hoch“ priorisierten Sicherheitsvorfällen durchgeführt. Bei einer *selektiven Datensicherung* werden nur einzelne ausgewählte Daten gesichert. Dieses Verfahren hat den Vorteil eines geringen Zeit- und Speicherplatzbedarfes. Nachteilig hingegen ist die genaue Festlegung, was gesichert werden soll und Verzicht auf nachträgliche Datenträgeranalyse.

– Bei Vorfällen mit geringer Priorität können „Vor-Ort-Untersuchungen“ mit Hilfe der mitgelieferten Werkzeuge ausgeführt werden (sollte vermieden werden). Da Speicherplatz zum Teil netzwerkseitig zur Verfügung gestellt wird, müsste, falls Netzverbindungen getrennt wurden, eine erneute Anbindung an das betroffene System erwogen werden.

System ist ausgeschaltet

- Bei einem ausgeschalteten physischen System beschränkt sich der Prozess auf das Sammeln verschiedenster Systeminformationen (Eventlogs, Konfigurationsdaten, etc). Hierfür soll das System von den zuständigen Systemadministratoren mittels einer Live-CD in einer forensische Umgebung gebootet werden. Auch hier gilt der Grundsatz, dass die gesammelten Daten auf dem zentralen Forensik-Server unter Beachtung forensischer Gesichtspunkte (Berechnung der Prüfsummen, Dokumentation, das Vier-Augen-Prinzip) gesichert werden.
- Bei $\geq$ „Hoch“ priorisierten Sicherheitsvorfällen kann ein forensisches Duplikat [5.8] angefertigt werden.

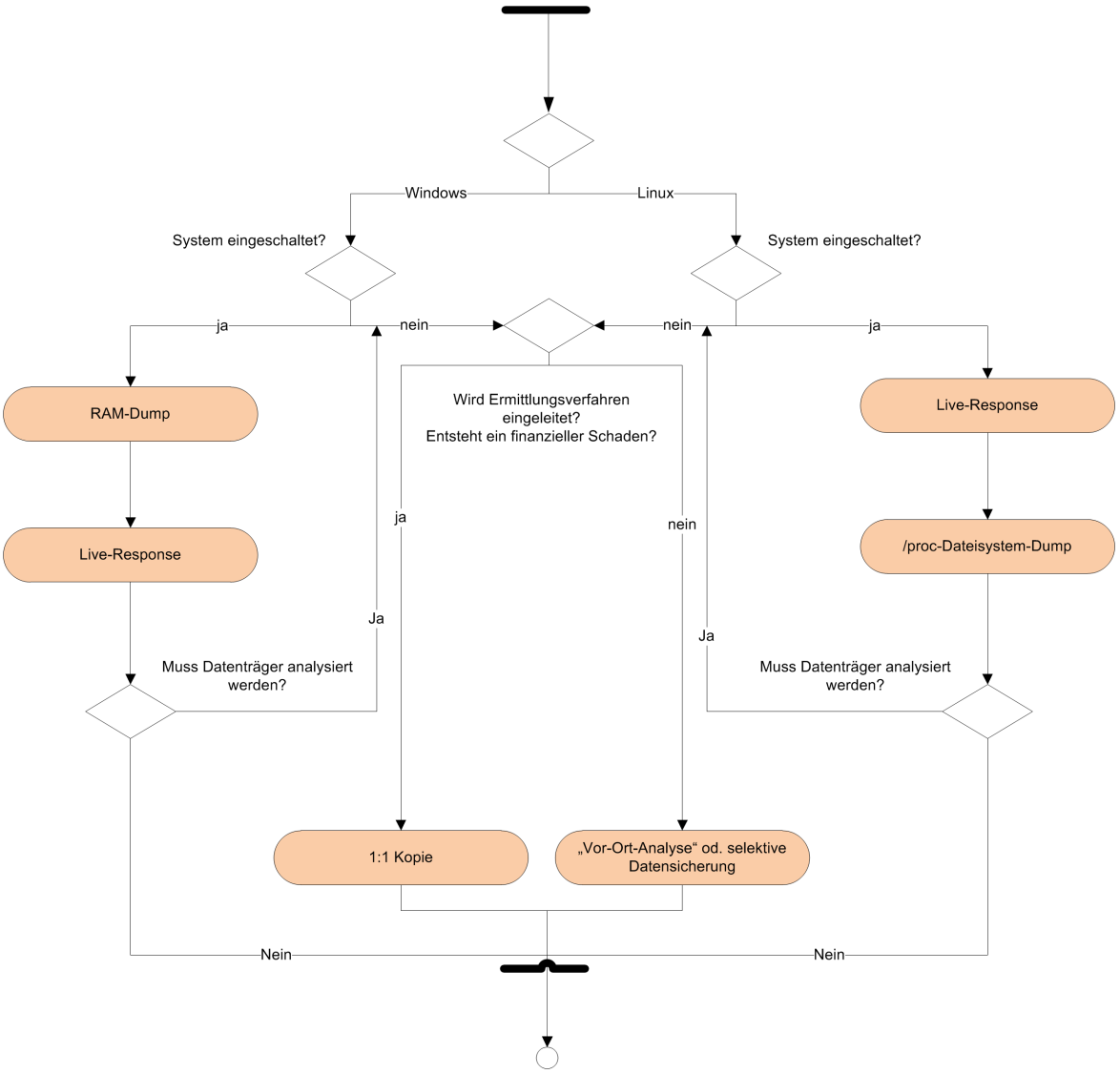

Abbildung 5.1.: Live-Response Prozessablauf bei physischen Maschinen

5.4.2. Virtuelles System

Virtualisierung stellt neue Herausforderungen an das Leibniz-Rechenzentrum, eröffnet aber gleichzeitig neue Möglichkeiten bei der forensischen Datenerfassung. In aller Regel kann das Sammeln aller relevanten Daten mit der Prozessbeschreibung für physische Maschinen erfolgen.

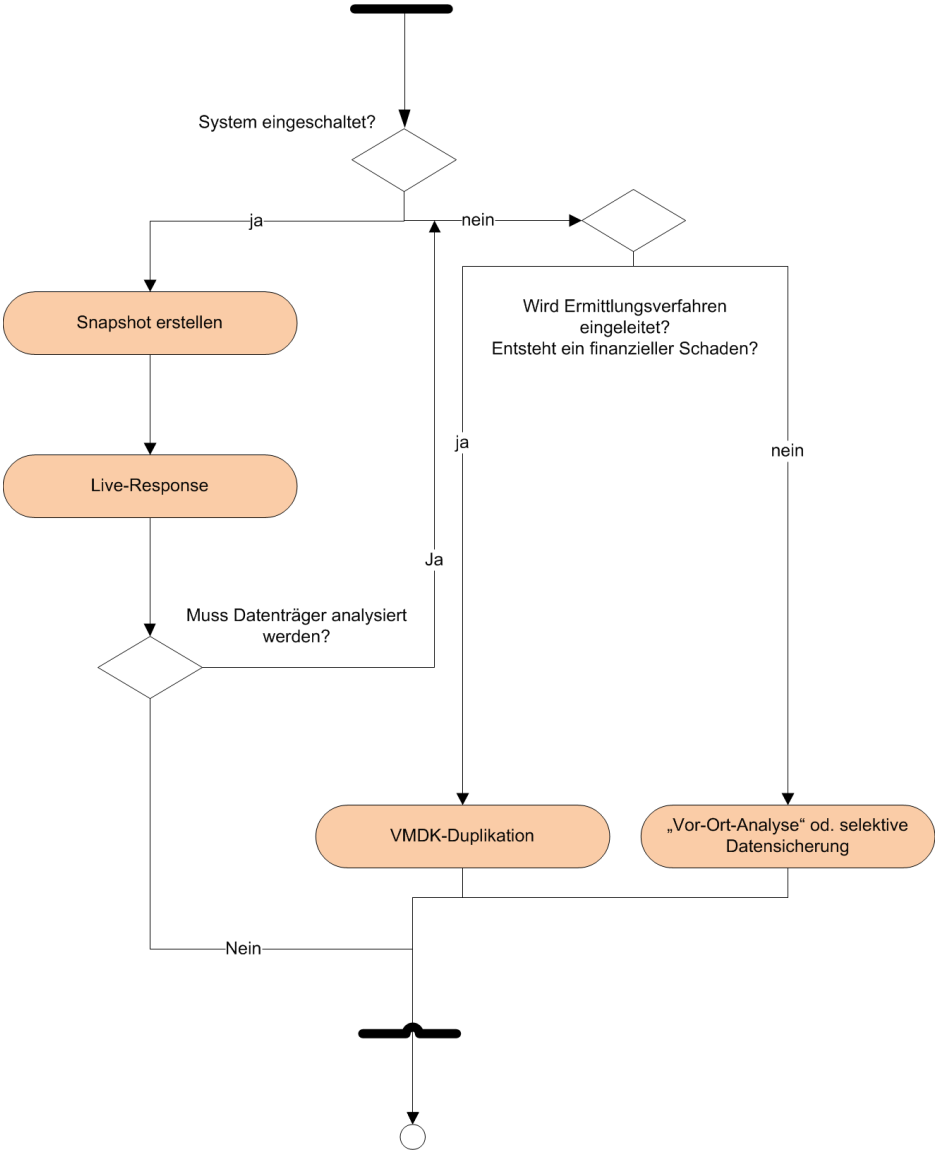

Abbildung 5.2.: Live-Response Prozessablauf bei virtuellen Maschinen

System läuft noch

Datensammeln bei virtuellen Systemen erfolgt fast nach dem gleichen Prinzip wie bei physischen Maschinen. Unterschiede werden in der Abbildung 5.2 aufgezeigt und nachfolgend dargestellt.

- Bevor mit der Datensammlung begonnen wird, sollte ein Snapshot des betroffenen Systems angefertigt werden. Sofern dabei Inhalt des Hauptspeichers in die Datei mit der Endung VMEM gesichert werden, kann die Erstellung eines RAM-Dumps übersprungen werden. Gängige Tools zur Hauptspeicheranalyse können VMEM-Dateien lesen und analysieren. Bei der Übertragung auf den zentralen Forensikserver müssen zwingend forensische Richtlinien beachtet werden, d.h. umfassende Dokumentation und Be-

rechnung der Prüfsummen. Erstellung eines Snapshots bedeutet gleichzeitig, dass das generelle Vorgehen sowohl bei Windows- als auch bei Linux-basierten Systemen von der Methodik her gleich ist.

- Auch hier wird mit den gleichen, vertrauenswürdigen Werkzeugen gearbeitet. Die Toolsammlung kann auf folgende Weise eingebunden werden:
  - Laden von einer ISO-Datei (das Einhängen kann nur durch einen VMware-Administrator erfolgen),
  - Laden von einem schreibgeschützten Network-Share.
- Nach dem Abschluss der Datensammlung wird das System mittels des VMware-Menüpunkts „Ausschalten“ heruntergefahren - dies entspricht dem sprichwörtlichen „Stecker ziehen“. Auf keinen Fall darf das VM ordnungsgemäß heruntergefahren werden.
- Ein forensisches Duplikat muss nicht explizit angefertigt werden. Stattdessen können VMDK-Dateien auf das Analysesystem übertragen und analysiert werden. Dieser Vorgang muss durch VMware-Administratoren angestoßen werden.

System ist ausgeschaltet

- Falls das VM bereits heruntergefahren wurde, beschränkt sich die Ermittlungstätigkeit auf die Sicherstellung der Systeminformationen.
- Bei $\geq$ „Hoch“ priorisierten Sicherheitsvorfällen können VMDK-Dateien sichergestellt werden.

Eine Anmerkung zum Schluss: der Einsatz von Hauptspeicheranalyse gewinnt seit 2006 immer weiter an Bedeutung und zählt mittlerweile zu den grundlegenden Maßnahmen der Computer-Forensik. In der Praxis wird sich zeigen, ob mit fortscheitender Erfahrung der zuständigen Ermittler, forensische Analysen bei virtuellen Maschinen auf die Hauptspeicher- sowie Logdateien-Analyse reduziert werden können. Die Vorteile liegen auf der Hand: mittels der Snapshot-Funktionalität von VMware sind RAM-Dumps in Handumdrehen angelegt und auf den zentralen Server übertragen. Das gleiche gilt für forensische Kopien der Datenträger im VMDK-Format. All das würde zu schnellen und effizienten Prozess-Schritten führen.

5.5. Datenerfassung unter Windows Server 2008R2

Im folgenden Abschnitt werden ausgewählte Tools, Ansätze und Methoden für die Durchführung der Live-Response an einem Windows-Server-2008-System vorgestellt.

5.5.1. Live-Response-Toolkit

Ganz bewusst wird im Folgenden so weit wie möglich auf die Verwendung der systemeigenen Befehle für die Sammlung der flüchtigen Informationen verzichtet, da das Risiko, ein trojanisiertes Programm zu verwenden, groß ist. Die unterschiedlichsten Schädlingsarten, die sich derzeit im Umlauf befinden, könnten Schadfunktionen aktivieren oder weitere Komponenten aus dem Internet nachladen. Zieht man die Locard’sche Regel in die Überlegungen mit ein, ist festzustellen, dass die Verwendung von Systembefehlen den Zeitstempel ihres letzten Aufrufs verändern würde - dies ist für eine forensische Untersuchung wenig wünschenswert.

Der Ansatz dieses Leitfadens ist auf die Verwendung von ausgewählten, vertrauenswürdigen Programmen ausgerichtet. Es handelt sich um eine Reihe von CLI-Tools[2]. Das erklärte Ziel: geringes Memory-Footprint bei wenigen Abhängigkeiten von externen Bibliotheken. Für Untersuchungen von virtuellen Maschinen wird eine ISO-Image-Datei vom SIR-Team gepflegt, für physische IT-Systeme sind relevante Tools auf einem schreibgeschützten USB-Stick abgelegt. Dabei kommen unter anderem folgende Tools zum Einsatz, die die benötigten Informationen [5.5.2] sehr schnell erfassen können [5.1]:

[2] Kommandozeilen-Tools

Tool	Beschreibung	Quelle
cmd.exe	leistungsfähiger Kommandozeileninterpreter für Windows Server 2008R2.	Microsoft
PsLoggedon	hilfreiches Tool zur Überwachung von lokal & remote angemeldeten Benutzern sowie zur Untersuchung von unberechtigten Zugriffen im Netzwerk	Sysinternals
LogonSessions	leistungsfähiges Befehlszeilentool, das alle angemeldeten Sitzungen auf einem Computer anzeigt.	Sysinternals
pskill	mit dem Befehl *pskill* lassen sich laufende Prozesse beenden.	Sysinternals
PsInfo	ein Befehlszeilenprogramm, das Informationen über IT-Systeme sammelt, u.a. Installationspfad, den Kernelbuild, Anzahl der Prozessoren und deren Typ, RAM-Größe, usw.	Sysinternals
PsService	hilfreiches Tool, um Systemdienste lokal oder auf Computern im Netzwerk anzuzeigen, zu beenden und zu starten.	Sysinternals
psfile	leistungsfähiges Tool zur Auflistung aller geöffneten freigegebenen Dateien im Netzwerk.	Sysinternals
psloglist	kompaktes Tool, um die Ereignisanzeigen verschiedener Computer einzusammeln, anzuzeigen und zu vergleichen.	Sysinternals
Tcpvcon	zeigt eine detaillierte Auflistung aller TCP- und UDP-Endpunkte an. Zusätzlich sind Prozesse, die auf die Endpunkte und Ports zugreifen, sichtbar.	Sysinternals
Listdlls	mit dem Befehl *Listdlls* lassen sich alle geladenen DLL-Dateien anzeigen.	Sysinternals
Handle	wichtiges Tool für die Analyse der laufenden Prozesse auf einem IT-System.	Sysinternals
NTLast	ein Befehlszeilenprogramm für die Anzeige von Eventlogs.	FoundStone
SFind	leistungsfähiges Tool, um die Alternate Data Streams auf NTFS-Systemen anzuzeigen. Bestandteil des Forensic Toolkits v2.0.	FoundStone
AFind	hilfreiches Tool, um MAC-Time auszulesen, ohne dabei Last Access Time zu verändern. Bestandteil des Forensic Toolkits v2.0.	FoundStone
MD5	ein Kommandozeilen-Programm zur Berechnung kryptographischer Hash-Funktion MD5.	fourmilab
windd	kompaktes Tool zur Erstellung eines Abbilds des Arbeitsspeichers.	moonsols
PromiscDetect	kompaktes Tool zur Überprüfung, ob ein Netzadapter im sog. Promiscuous-Modus arbeitet und Informationen über ankommenden und abgehenden Datenverkehr an einen „Sniffer“ weiterleitet.	ntsecurity

Tlist	kompaktes Tool zur Auflistung von detaillierten Informationen über einzelne Prozesse, Anzeige von vollständigen Pfaden und Aufrufparametern eines Prozesses möglich.	Microsoft
reg	ein Kommandozeilenprogramm zur selektiven Sicherung der Windows-Registrierungsdatenbank.	Microsoft
regdump	ein Befehlszeilentool zur vollständigen Sicherung der Registry.	Microsoft
arp	ein hilfreiches Tool zur Auflistung der Übersetzungstabellen für IP-Adressen, die von Address Resolution Protocol verwendet werden.	Microsoft
route	mit dem Befehlt `route` lassen sich aktuelle Routing-Tabellen anzeigen.	Microsoft
nbtstat	ein Kommandozeilenprogramm zur Anzeige der Protokollstatistik und der aktuellen Verbindungen, die NetBIOS über TCP/IP verwenden.	Microsoft
netstat	wichtiges Programm zum Abrufen der detaillierten Protokollstatistiken und zur Anzeige aller aktiven TCP-, UDP- und IP-Verbindungen.	Microsoft
ipconfig	ein Kommandozeilentool, das Informationen über verwendete Netzwerkadapter anzeigt.	Microsoft
auditpol	mit dem Kommandozeilenbefehl *auditpol* lassen sich die aktuellen Richtlinien für angemeldete Benuter auflisten.	Microsoft

Tabelle 5.1.: Bestandteile des Windows Server 2008R2 Response Toolkits.

5.5.2. Live-Response

Wie bereits in Abschnitt 5.1 beschrieben sind bei der Live-Response verschiedene Arten von Daten für den Fortgang der Ermittlung von Interesse. Kern der Datenerfassung bildet ein Top-Down-Ansatz für die Bestimmung der Reihenfolge der Datensammlung. Auch wenn die Reihenfolge der Sicherung der flüchtigen Daten vom konkreten Untersuchungsfall abhängt, reicht in der Regel folgende Priorisierung der Tätigkeiten für bekannte Sicherheitsvorfälle aus - unabhängig davon, ob es sich um eine virtuelle oder eine physische Maschine handelt. Eine alternative Vorgehensweise wird am Ende des Kapitels vorgestellt. Typischerweise geht man in folgender Reihenfolge vor:

1. vertrauenswürdige *cmd.exe* aufrufen
2. aktuelle Systemzeit und deren Differenz zu einer vertrauenswürdigen Referenzzeit erfassen
3. Sicherung des Hauptspeicherinhalts
4. Informationen über eingeloggte Benutzer sichern
5. Informationen über Netzverbindungen (offene Ports, aktive Verbindungen, gerade geschlossene Verbindungen, ...) speichern
6. Sicherung der Informationen über laufende Prozesse (Mutterprozesse, Prozesseigentümer, Prozessorzeiten, Pfad, Aufrufparameter, ...)
7. aktuelles Portmapping ermitteln
8. Informationen über aktuellen Netzstatus bestimmen
9. Eventlogs sichern

10. Registry sichern
11. aktuelle Systemzeit erfassen
12. Ergebnisse der Untersuchung sichern

Bei der Auswahl der Methodik und Hilfsmittel wurde darauf geachtet, dass sie in der Fachwelt allgemein anerkannt und beschrieben worden sind - eine wichtige Voraussetzung, damit die gewählte Vorgehensweise vor Gericht Bestand hat (vgl. 4.2).

5.5.2.1. Aktuelle Systemzeit erfassen

Die Erfassung des aktuellen Datums und - insbesondere - der aktuellen Uhrzeit ist notwendig, um deren Differenz zu einer vertrauenswürdigen Referenzzeit zu erfassen, Systemereignisse zu korrelieren und die zur Ermittlungszwecken durchgeführten Tätigkeiten zeitlich einordnen zu können. Beide Befehle *date* und *time* sind fester Bestandteil der cmd.exe-Anwendung. Abbildung 5.3 zeigt Beispiele für Ausführung der Kommandos und die Umleitung der Ausgabe in eine Text-Datei.

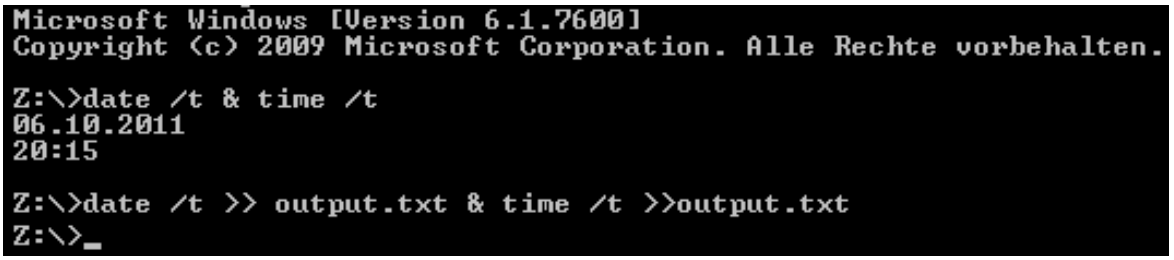

Abbildung 5.3.: Erfassung der aktuellen Systemzeit/des aktuellen Datums unter Win Server 2008R2

5.5.2.2. Arbeitsspeicherabbild erstellen

Um zuverlässige Aussagen über den Zustand eines laufenden Systems treffen zu können, wird in einigen Fällen ein vollständiges Hauptspeicherabbild in eine Datei gesichert - über Notwendigkeit wird bei jedem Sicherheitsvorfall individuell entschieden [5.4.1]. Die Vorteile liegen auf der Hand: aus dem Hauptspeicher-Image lassen sich beispielsweise Verwaltungsinformationen zu Prozessen und Threads herauslesen sowie der von den laufenden Prozessen geladenen DLLs und geöffneten Dateien. Unter Umständen lassen sich auch bereits terminierte Prozesse und Netzwerkverbindungen ebenfalls nachweisen (sofern die betreffenden Speicherbereiche noch nicht überschrieben wurden). Auch wird die komplette Windows-Registry im Hauptspeicher gehalten und lässt sich somit bequem nach Hinweisen durchsuchen. All dies geschieht off-site, das eigentliche System wird dabei nicht verändert und das Windows-API, das oft von Schadcode trojanisiert ist, wird dabei umgangen.

In letzter Zeit sind zahlreiche Methoden und Werkzeuge für die Erstellung eines Hauptspeicherabbildes entwickelt worden. Zwei Tools werden in diesem Unterkapitel vorgestellt und die technischen Gegebenheiten erklärt. VMware ESXi Virtualisierungslösung bietet außerdem eine weitere Möglichkeit ein Arbeitsspeicherabbild schnell und ohne Einsatz von externen Tools zu erstellen.

MoonSols Windows Memory Toolkit MoonSols Windows Memory Toolkit ist eine kostenfreie Sammlung von Kommandozeilenwerkzeugen, entwickelt von Matthieu Suiche. Neben diversen Tools für die Umwandlung von Speicherabbildern sind zwei Tools *win32dd* (für 32-Bit Systeme) und *win64dd* (für 64-Bit Systeme) für die RAM-Aquisition enthalten. Beide Werkzeuge nutzen Windows-Kernel-Gerätetreiber, um aus dem Kernelmodus ein Abbild des physikalischen Speichers zu erstellen. Das Programm lässt sich über einige Parameter steuern, die entweder beim Aufruf in der Kommandozeile oder über eine Batch-Datei übergeben werden können. Mögliche Aufrufparameter sind der Abbildung 5.4 zu entnehmen.

```
Z:\>win64dd -h
  win64dd - 1.3.1.20100417 - (Community Edition)
  Kernel land physical memory acquisition
  Copyright (C) 2007 - 2010, Matthieu Suiche <http://www.msuiche.net>
  Copyright (C) 2009 - 2010, MoonSols <http://www.moonsols.com>

Usage: win64dd [options]

  Option        Description
  ------        -----------
  /f <file>     File destination.

  /r            Create a Raw memory dump file. (default)

  /d            Create a Microsoft memory crash dump file. (WinDbg compliant, XP and later only)

  /c <value>    Memory content.
                   0 - Full physical address space.
                   1 - Memory manager physical memory block. (default)
                   2 - Memory manager physical memory block + Very First PFNs.

  /m <value>    Mapping method for either /d or /r option.
                   0 - MmMapIoSpace().
                   1 - \\Device\\PhysicalMemory.
                   2 - PFN Mapping. (default)

  /e            Create a Microsoft hibernation file. (local only, reboot)

  /k            Create a Microsoft memory crash dump file (BSOD).
                (local only, reboot)

  /s <value>    Hash function to use. (Only on sender/local machine)
                   0 - No hashing algorithm. (default)
                   1 - SHA1 algorithm.
                   2 - MD5 algorithm.
                   3 - SHA-256 algorithm.

  /y <value>    Speed level.
                   0 - Normal.
                   1 - Fast.
                   2 - Sonic.
                   3 - Hyper sonic. (default)

  /t <addr>     Remote host or address IP.
  /p <port>     Port, can be used with both /t and /l options. (default: 1337)

  /l            Server mode to receive memory dump remotely.

  /a            Answer "yes" to all questions. Must be used for piped-report.

  /?            Display this help.

Samples:
  win64dd /d /f physmem.dmp               - Standard Microsoft crash dump.

  win64dd /m 0 /r /f F:\physmem.bin       - Raw dump using MmMapIoSpace() method.


  win64dd /l /f F:\msuiche.bin            - Waiting for a local connexion on port 1337.
  win64dd /t sample.foo.com /d /c 0       - Send remotely a Microsoft full crash dump.

  win64dd /d /f \\smb_server\remote.dmp   - Send remotely on a SMB server.
```

Abbildung 5.4.: win64dd - Beispielausgabe

Abhängig vom Standort des infizierten Systems kann es erforderlich sein, Memory Dump direkt übers Netzwerk an das Analysesystem zu schicken und dort abzulegen. WinDD bringt hierfür eigene Client- und Serverfunktionalität mit. Zunächst ist es erforderlich den Server auf dem Analysesystem mit dem Befehl *win64dd.exe /l /f memdump.dmp* zu initialisieren, um die Verbindung vorzubereiten. Anschließend muss der zugehörige Client auf dem betroffenen IT-System lediglich mit dem Parameter */t 'IP-Adresse des Analysesystems'* aufgerufen werden - die Verbindung wird hergestellt und die Erstellung des Hauptspeicherabbilds gestartet.

Neben der kostenlosen Community-Edition wird vom Hersteller noch die Professional-Edition angeboten. Wichtigste Unterschiede umfassen die Möglichkeit die Hybernationabbilder ins Microsoft Windows Debugger Format [Micb] zu konvertieren und VMware-Snapshots auszulesen.

Guidance Software WinEn WinEn ist eine Standalone-Applikation zur Erstellung der Speicherabbilder, die ab der Version 6.11 der Forensiksoftware EnCase ausgeliefert wird. Es läuft als Texttool in der Eingabeaufforderung von Windows. Beim Aufruf von WinEn ohne weitere Parameter werden einige Daten abgefragt, die sich alternativ als Argumente übergeben oder in einer Konfigurationsdatei festlegen lassen. Um die professionelle und nachvollziehbare Analyse zu ermöglichen, speichert WinEn RAM-Dumps im propriäteren Expert

Witness-Format (.E01), welches dann in EnCase geladen und analysiert werden kann (eine spätere Umwandlung in ein reines Speicherabbild-Format ist möglich) - es werden einige Meta-Daten wie zum Beispiel der Name des Bearbeiters und ein Fallname gespeichert:

```
Z:\>winen.exe
Please enter a value for the option \"EvidencePath\":
 Z:\memdump
Please enter a value for the option \"EvidenceName\":
 ram
Please enter a value for the option \"CaseNumber\":
 001
Please enter a value for the option \"Examiner\":
 Metzger
Please enter a value for the option \"EvidenceNumber\":
 001
Please enter a value for the option \"Compress\":
 2
```

Listing 5.1: WinEn

Die Daten lassen sich komprimieren, wobei eine Datenreduktion um bis zu 50% erreicht werden kann.

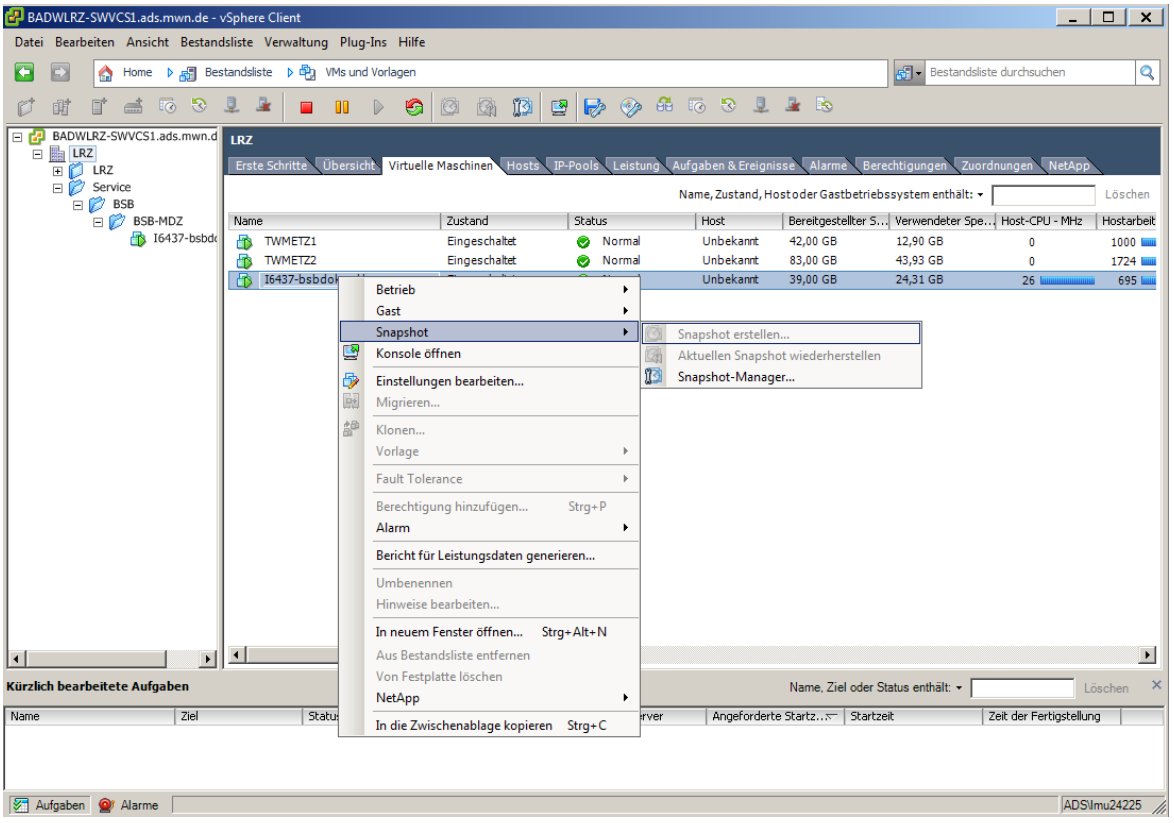

Abbildung 5.5.: ESXi - System kann in den Suspend-Modus versetzt werden.

VMware Snapshots Eine Alternative zu den vorgestellten Softwarelösungen ist die VMware-Funktion „Snapshot erstellen...“ [5.5] - ESXi verfügt über einen Mechanismus, der Daten aus dem Arbeitsspeicher in eine Swap-Datei mit der Endung .vmem auslagert. Diese Funktion ist vor allem für das Leibniz-Rechenzentrum interessant, immerhin steigt die Anzahl der virtualisierten Systeme kontinuierlich an. Die VMEM-Datei weist keine relevanten Unterschiede zu einem mit konventionellen Tools erstellten Arbeitsspeicherabbild [Car09]

und kann mit den gängingen Memory-Analyse-Programmen (z.B. Volatility Framework oder HBGary) ausgelesen und untersucht werden. Wesentlicher Vorteil dieser Vorgehensweise: eine VMEM-Datei ist mit einigen wenigen Aktionen erstellt und kann mit geeigneten Tools (z.B. dd inklusive Berechnung eines Hash-Wertes) auf das Analysesystem übertragen werden - der Effizienzgewinn ist nicht zu vernachlässigen. Ein weiterer Vorteil: zur Standardvorgehensweise am LRZ gehört das Kappen der Netzverbindungen, oftmals bevor Live-Response durchgeführt werden kann - Gefahren, die isoliert werden, können immerhin keinen Schaden anrichten. Erstellt man ein Snapshot des kompromittierten VMs bleiben alle wichtige Daten für die spätere Analyse erhalten [DiMb].

5.5.2.3. Informationen über eingeloggte Benutzer sichern

Der nächste Schritt besteht darin, Hinweise über eingeloggte Benutzer zu erhalten - insbesondere Informationen über remote eingeloggte Benutzer sind für die weitere Ermittlung von wesentlichem Interesse (wenn auch Innentäter keineswegs unterschätzt werden sollten). Um alle angemeldeten Benutzer anzuzeigen, kommt das bewährte Tool von Mark Russinovich *PsLoggedOn* in der zum Zeitpunkt der Erstellung dieser Arbeit aktuellen Version 1.34 zum Einsatz. Das Kommandozeilenprogramm durchsucht zunächst den Windows Registrierungsdatenbank Zweig *HKEY_USERS* nach hinterlegten *SIDs* (security Identifier) und zeigt derzeit aktive User an. Um remote angemeldete Benutzer zu ermitteln, wertet PsLoggedOn die NetSessionEnum-API ([Sysa], vgl. Abb. 5.6).

```
Z:\>psloggedon

PsLoggedon v1.34 - See who's logged on
Copyright (C) 2000-2010 Mark Russinovich
Sysinternals - www.sysinternals.com

Users logged on locally:
     17.09.2011 10:56:00        Tolwyn-PC\Tolwyn

No one is logged on via resource shares.
```

Abbildung 5.6.: PsLoggedOn in Aktion

```
Z:\>logonsessions -p

Logonsesions v1.21
Copyright (C) 2004-2010 Bryce Cogswell and Mark Russinovich
Sysinternals - www.sysinternals.com


[0] Logon session 00000000:000003e7:
    User name:    WORKGROUP\TOLWYN-PC$
    Auth package: NTLM
    Logon type:   (none)
    Session:      0
    Sid:          S-1-5-18
    Logon time:   17.09.2011 10:55:35
    Logon server:
    DNS Domain:
    UPN:
      404: smss.exe
      628: csrss.exe
      688: wininit.exe
      712: csrss.exe
      748: services.exe
      768: lsass.exe
      776: lsm.exe
      836: winlogon.exe
      936: svchost.exe
      452: cmdagent.exe
     1000: atiesrxx.exe
     1076: svchost.exe
     1120: svchost.exe
```

Abbildung 5.7.: LogonSessions zeigt ausführliche Informationen über aktive Anmeldesitzungen

Um an ausführlichere Informationen zu kommen, wird ein weiteres Befehlszeilentool von Sysinternals eingesetzt - *LogonSessions*. Mit dem von Bryce Cogswell und Mark Russinovic entwickelten Programm erhalten

die Ermittler schnell und unkompliziert sehr ausführliche Informationen über aktive Anmeldesitzungen und, bei Verwendung des zusätzlichen Parameters *-p*, die in jeder Sitzung ausgeführten Prozesse. Neben der aktiven Benutzern werden auch die Systemkonten angezeigt.

Moderne Schadsoftware verwendet zur Tarnung verschiedene Evasion-Techniken, z.B. Authentifizierung über rohen TCP-Datenstrom unter Umgehung der Windows-Authentifizierung (NTLM) und tauchen daher nicht in der Liste der aktiven Benutzer auf. Allerdings kann man sich die Ergebnisse der Ausgabe trotzdem zunutze machen - würde man zu einem späteren Zeitpunkt einen nicht in der Liste aufgeführten Nutzer entdecken, ist es ein deutlicher Hinweis, dass sich ein Rootkit auf dem System eingenistet hat.

Finden sich auf einem verdächtigen System dennoch Hinweise auf remote eingeloggte Benutzer, liefert ein weiteres SysInternals-Tool Hinweise auf Aktivitäten des Nutzers. Mit dem Werkzeug *PsFile* lassen sich alle geöffneten Dateien anzeigen, die über Dateifreigaben im Netzwerk verteilt werden [5.8].

```
Z:\>psfile

psfile v1.02 - psfile
Copyright © 2001 Mark Russinovich
Sysinternals

Files opened remotely on TOLWYN-PC:

[47] S:\CGI\trunk\3dsmax\output\episode_01_outro\scene_01_0001.png
    User:   TOLWYN
    Locks:  0
    Access: Read
[48] S:\CGI\trunk\3dsmax\output\episode_01_outro\scene_01_0000.png
    User:   TOLWYN
    Locks:  0
    Access: Read
```

Abbildung 5.8.: PsFile zeigt alle geöffneten freigegebenen Dateien im Netzwerk an

5.5.2.4. Informationen über Netzverbindungen speichern

Betrachtet man die typische Vorgehensweise der Angreifer auf Windows-Systemen, ist festzustellen, dass der Inhalt der NetBIOS-Tabelle mit dem Befehl *nbtstat -c* ausgelesen wird. Kenntnis über die lokale Netzumgebung nutzt ein Hacker aus, um weitere angreifbare Systeme zu lokalisieren und zu kompromitieren. Diese Systeme sollte man ebenfalls im Auge behalten, um sicherzugehen, dass diese vom Sicherheitsvorfall nicht betroffen waren und dass ein Angreifer keine Manipulationen wie trojanische Pferde hinterlassen hat.

Unabhängig davon sollten alle aktiven Netzverbindungen gesichert werden - es lassen sich wichtige Anhaltspunkte über eventuell aktive Hintertürprogramme in der Auflistung der offenen Verbindungen bzw. Netzwerkports. Es sollte damit gerechnet werden, dass bestehenden Netzverbindungen nach einem Timeout geschlossen werden und sich nicht mehr nachweisen lassen. Informationen vom betroffenen System könnten sehr wertvolle Hinweise liefern, die für das weitere Vorgehen bei der Live Response sehr hilfreich sein kann: der Angreifer könnte noch auf dem System aktiv sein oder ein installierter Trojaner könnte Verbindung zu einem Command & Control Server (C&C) aufnehmen oder weiteren Schadcode aus dem Internet nachladen.

Für die Analyse der ein- und ausgehenden Verbindungen kommt das bekannte CLI-Programm *netstat* zum Einsatz. Um die professionelle und umfassende Analyse zu ermöglichen, verfügt netstat über diverse Aufrufparameter. Die übliche Vorgehensweise für die Auswertung umfasst das Aufrufen von netstat mit dem Befehlszeilenparamter *-ano*. Dies gestaltet dem untersuchenden Ermittler eine Auflistung aller TCP- und UDP-Verbindungen, ihren Status sowie verwendete Ports und Process identifier (PID) anzuzeigen [5.9]. Dieser Ansatz ist sinnvoll um anormale Verbindungen zu identifizieren: finden sich z.B. sehr viele Verbindungsaufbauanfragen, ist davon auszugehen, dass das IT-System für einen Portscan verwendet wurde. Die Tarnmechanismen der Malware sind jedoch so zahlreich, dass ohne weitere Nachforschungen die von der Schadsoftware iniziierten Connections für die Ermittler (und IDS) wie legitime Verbindungen aussehen.

```
Z:\>netstat -ano

Aktive Verbindungen

  Proto  Lokale Adresse          Remoteadresse          Status          PID
  TCP    0.0.0.0:80             0.0.0.0:0              ABHÖREN         2032
  TCP    0.0.0.0:135            0.0.0.0:0              ABHÖREN         1016
  TCP    0.0.0.0:443            0.0.0.0:0              ABHÖREN         2032
  TCP    0.0.0.0:445            0.0.0.0:0              ABHÖREN         4
  TCP    192.168.0.16:58490     212.227.17.186:993     HERGESTELLT     4580
  TCP    192.168.0.16:58499     199.59.148.139:443     HERGESTELLT     9156
  TCP    192.168.0.16:58500     199.59.148.139:443     HERGESTELLT     9156
  TCP    192.168.0.16:58606     199.59.148.139:443     HERGESTELLT     9156
  TCP    192.168.0.16:58632     199.59.148.139:443     HERGESTELLT     9156
```

Abbildung 5.9.: Netstat zeigt alle geöffneten TCP-/UDP-Verbindungen an

Diese Informationen sind zudem eng verbunden mit den Routingtabellen. Netstat bietet eine zusätzliche Methode für die Anzeige von allen aktiven Routen - sie wird bei Bedarf mit dem Parameter *-anor* aktiviert. Dies gestaltet für den Forensiker die Netzkonfiguration anschaulicher und bietet den IT-Administratoren gleichzeitig die Möglichkeit, Ansätze für diverse Problemlösungen zu finden.

5.5.2.5. Sicherung der Informationen über laufende Prozesse

Bei jeder forensischen Analyse ist es wichtig, eine Liste der aktuell laufenden Prozesse zu sichern. Hierzu gehören nicht nur die Auflistungen aller aktiven Prozesse, sondern auch weitere Informationen, die aus dem Windows-Taskmanager nicht (oder nur teilweise) herausgelesen werden können:

- vollständiger Pfad zur ausführbaren Datei,
- Aufrufparameter (sofern verwendet),
- Prozesslaufzeit,
- Prozesseigentümer,
- geladene Module / Bibliotheken,
- zugehörige Speicherbereiche.

Vor allem der vollständige Pfad zur ausführbaren Datei und verwendete Aufrufparameter sind für die Behandlungs von Sicherheitsvorfällen von großem Wert. Es sollte damit gerechnet werden, dass eingeschleuste Schadsoftware sich als bekannter Windows-Prozess - beispielsweise *Svchost.exe* - tarnt. *Svchost.exe* startet beim Hochfahren des IT-Systems die in der Windows-Registry eingetragene DLL-Dateien als Dienste, es können mehrere Instanzen von *Svchost.exe* gleichzeitig ausgeführt werden. Für eine Firewall kann solch ein Prozess nach außen hin alle Kriterien der festgelegten Sicherheitsregeln erfüllen und wird deshalb auch durchgelassen.

Eine einfache Möglichkeit Schadprozesse aufzuspüren ist die Analyse des Dateipfades. Die Datei *svchost.exe* befindet sich im Ordner *C:\Windows\System32*. Wenn das nicht der Fall ist, handelt es sich bei *svchost.exe* um einen Schädling. Der Wurm *W32/Nachi* platziert zum Beispiel eine *svchost.exe* Datei im *C:\Windows \system32\Wins* Verzeichnis - Tests zeigten, dass es sich hierbei um einen einfachen TFTP-Client handelte. Deshalb sollten die IT-Systeme, mit den entsprechenden Prozessüberwachungswerkzeugen analysiert werden - sie werden auf den folgenden Seiten vorgestellt. Allerdings sind dem Einfallsreichtum der Virenautoren keine Grenzen gesetzt - neuere Schädlinge, wie der Conficker Wurm, ändern die Windows-Registry, so dass die *svchost.exe* die Schädlings-DLL-Datei ausführt - und der Schadcode als vermeintlich regulärer Dienst gestartet wird.

Tlist Für die Verwaltung von Prozessen auf der Kommandozeilenebene stellt Microsoft ein Tool von der Windows 2000-CD zur Verfügung: *tlist*.

Durch den Befehl „*tlist.exe*“ wird eine Liste der aktiven Prozesse angezeigt. Das Programm lässt sich über einige Parameter steuern, die beim Aufruf eingegeben werden können. Durch die Befehlszeilenoption -v werden eine Reihe von nützlichen Informationen abgerufen: Session-ID, Prozess-ID, Prozessname, aktive Dienste sowie verwendete Aufrufparameter. Die folgende Beispielausgabe für *Tlist* zeigt eine ausgeführte Instanz von *Svchost.exe*:

```
[0]  0 64 3912 svchost.exe     Svcs:  stisvc
     Command Line: Z:\Windows\system32\svchost.exe -k imgsvc
```

Abbildung 5.10.: Tlist zeigt ausführliche Informationen zu Prozessen an

Andere Parameter geben diese Informationen einzeln aus: *-s* zeigt eine Liste der aktiven Dienste in den einzelnen Prozessen an, *-c* zeigt Kommandozeilenparameter, mit dem Argument *pid* erhält man Detailinformationen zum angegebenen Prozess (z.B. welche Bibliotheken vom Prozess geladen wurden und wieviel physischen Speicher der Prozess belegt), über den Schalter *-m* lässt sich herausfinden, welche Prozesse eine bestimmte DLL-Datei referenzieren. Schließlich lässt sich mit dem Parameter *-t* der komplette Prozessbaum [5.11] anzeigen - somit kann man leicht die Prozesskette erkennen.

```
atiesrxx.exe (1000)
  atieclxx.exe (1584) AMD EEU Client
svchost.exe (1032)
  audiodg.exe (9700)
svchost.exe (1076)
  WUDFHost.exe (4924)
  dwm.exe (5420)
svchost.exe (1120)
  taskeng.exe (2120)
    iDisplay.exe (2280)
      iDisplayX64Helper.exe (2924)
  wuauclt.exe (3148)
  taskeng.exe (4828)
    updaterstartuputility.exe (5568)
svchost.exe (1304)
spoolsv.exe (1616)
```

Abbildung 5.11.: Tlist zeigt vollständigen Prozessbaum

```
cmd.exe pid: 10648
Command line: "Z:\Windows\system32\cmd.exe"

Base                Size      Path
0x000000004a8b0000  0x58000   Z:\Windows\system32\cmd.exe
0x0000000077210000  0x1ac000  Z:\Windows\SYSTEM32\ntdll.dll
0x0000000076e40000  0x11f000  Z:\Windows\system32\kernel32.dll
0x00000000fd9a0000  0x6c000   Z:\Windows\system32\KERNELBASE.dll
0x00000000fe0a0000  0x9f000   Z:\Windows\system32\msvcrt.dll
0x00000000fc0c0000  0x8000    Z:\Windows\system32\WINBRAND.dll
0x0000000076b30000  0xfa000   Z:\Windows\system32\USER32.dll
0x00000000fe030000  0x67000   Z:\Windows\system32\GDI32.dll
0x00000000fe3d0000  0xe000    Z:\Windows\system32\LPK.dll
0x00000000ff450000  0xca000   Z:\Windows\system32\USP10.dll
0x00000000fdd60000  0x2e000   Z:\Windows\system32\IMM32.DLL
0x00000000fdf20000  0x109000  Z:\Windows\system32\MSCTF.dll
0x0000000080000000  0x5c000   Z:\Windows\system32\guard64.dll
0x00000000fdc30000  0xdb000   Z:\Windows\system32\ADVAPI32.dll
0x00000000fdd40000  0x1f000   Z:\Windows\SYSTEM32\sechost.dll
0x00000000ff1c0000  0x12e000  Z:\Windows\system32\RPCRT4.dll
0x00000000fd670000  0x9000    Z:\Windows\system32\fltlib.dll
0x00000000fd540000  0x57000   Z:\Windows\system32\apphelp.dll
```

Abbildung 5.12.: ListDLLs zeigt alle DLLs, die von gerade laufenden Programmen verwenden werden.

ListDLLs Bei den Ermittlungen stehen die laufenden Prozessen und die von ihnen geladenen dynamischen Verbindungsbibliotheken in einem direkten Verhältnis. Dieses Bewusstsein muss bei den Forensikern vorhanden sein, da mehr oder weniger alle gängige Programme auf die von DLLs bereitgestellten Funktionen zugreifen. Das Befehlszeilen-Tool *ListDlls* zeigt den kompletten Pfadnamen der DLL -Dateien an, die zum Ausführungszeitpunkt auf dem IT-System gestartet sind [5.12]. DLLs sind aber eben auch effektive Methode für erfolgreiche Angriffe - Hacker injizieren Code in den Adressraum eines anderen Prozesses und bringen ihn zur Ausführung. Für Ermittler bedeutet diese Tatsache zweierlei: Zum einen lässt sich mit der DLL-Injection Datentransfer als bekannte Anwendung getarnt an der Firewall vorbei schleusen, zum anderen taucht so ein Prozess nicht in der Prozessliste mangels eigener PID auf [Creb].

Handles Das kleine SysInternals-Programm *Handle* kann Informationen zu offenen Handles für alle Prozesse im System anzeigen. Das Dienstprogramm wertet, bei Verwendung des Befehlszeilenarguments *-a*, Verweise auf offene Dateien, Ports, Registrierungsschlüssel, Synchronisierungsprimitive, Threads und Prozesse aus - dank dieser Funktionen lassen sich von einem verdächtigen Prozess verwendete Bilbiotheken, Registrierungsdatenbankeinträge und dergleichen aufspüren. Mit verschiedenen Befehlszeilenparametern lässt sich die Ausgabe verfeinern. Abbildung 5.13 erlaubt einen Einblick in die Funktionsweise von Handle anhand von *cmd.exe*.

```
cmd.exe pid: 10648 Tolwyn-PC\Tolwyn
    4: Directory      \KnownDlls
    8: File  (RW-)   Z:\
    C: Event
   10: ALPC Port
   14: ALPC Port
   18: Key           HKLM\SYSTEM\ControlSet002\Control\Nls\Sorting\Versions
   1C: Key           HKLM\SYSTEM\ControlSet002\Control\SESSION MANAGER
   20: EtwRegistration
   24: Event
   28: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
   2C: Desktop       \Default
   30: WindowStation \Sessions\1\Windows\WindowStations\WinSta0
   34: File  (R-D)   Z:\Windows\System32\de-DE\cmd.exe.mui
   38: Key           HKLM
   3C: Thread
   40: Mutant
   44: Event
   48: File  (---)   \FileSystem\Filters\FltMgrMsg
   4C: Key           HKCU
   50: Key           HKLM\SYSTEM\ControlSet002\Control\Nls\Locale
   54: Key           HKLM\SYSTEM\ControlSet002\Control\Nls\Locale\Alternate Sorts
   58: Key           HKLM\SYSTEM\ControlSet002\Control\Nls\Language Groups
   6C: Key           HKCU\Software\Microsoft\Windows NT\CurrentVersion
   70: Key           HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags
   74: Key           HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options
```

Abbildung 5.13.: Handle listet laufende Prozesse und File-Handles

Die Auswahl des richtigen Tools ist schwierig und ist unbestritten ein Eckpfeiler bei einer forensischen Untersuchung. Abhängig vom konkreten Untersuchungsfall könnte eine einfache Prozessübersicht ausreichend sein, um die gesteckte Ziele zu erreichen. Es ist aber nicht auszuschließen, dass erst eine Kombination von mehreren Tools (z.B. *Tlist* mit dem *-x* Schalter in Verbund mit *ListDLLs* und *Handle*) den entscheidenden Hinweis liefert.

5.5.2.6. Aktuelles Portmapping ermitteln

Wesentliche Anhaltspunkte über eventuell aktive Hintertürprogramme finden sich auch in der Auflistung des aktuellen Prozess-zu-Ports-Mappings. Somit lassen sich Schadprogramme entlarven, die Daten übers Netz übermitteln. Es existieren mehrere, in der Funktionalität sehr ähnliche Programme um alle TCP- und UDP-Ports aufzulisten und deren Zustand und zugehörigen Prozessnamen aufzuzeigen: standardmäßig mit Windows mitgelieferte Tool *netstat*, *FPort* von McAfee (benötigt Administratorprivilegien für die Ausführung) und schließlich *Tcpvcon* von SysInternals. Von den drei Tools ist *Tcpvcon* am intuitivsten nutzbar [5.14]. Das Programm versucht Domainnamen aufzulösen und zu protokollieren. Bei Malware wird oft auf DynDNS[3] zurückgegriffen, langfristig gesehen sind die IP-Adressen in so einem Fall wertlos. Ferner lässt sich die Ausgabe im CSV-Format für eine spätere, tiefergehende Analyse und Korrelation exportieren. Zu bedenken ist,

[3] Beim so genannten DynDNS leitet ein Server Domain-Anfragen an eine dynamische IP-Adresse weiter.

dass alle an dieser Stelle erwähnten Tools auf das System-API zurückgreifen, um die Informationen über offene Verbindungen auszulesen.

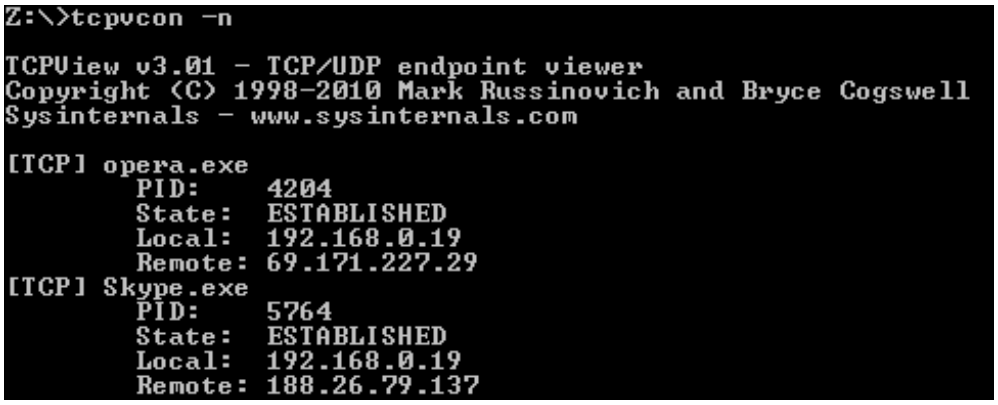

Abbildung 5.14.: Tcpvcon: Alle Ports auf einen Blick

5.5.2.7. Informationen über aktuellen Netzstatus bestimmen

Es sollte damit gerechnet werden, dass der Angreifer die Netzkonfiguration der betroffenen Systeme manipulieren könnte. Deshalb sollten die Ermittler die Konfigurationseinstellungen mit den entsprechenden Tools auslesen. Neben dem Windows-Programm *ipconfig* - einem Eingabeaufforderungstool, welches folgende Informationen im erweiterten Modus anzeigen kann: Hostname, DNS-Server, NetBIOS-Knotentyp, NetBIOS-Bereichs-ID, IP-Routing, WINS-Proxy, NetBIOS-Auflösung durch DNS, etc - könnten beispielsweise Tools zum Einsatz kommen, die darauf spezialisiert sind Netzadapter im Promiscious-Modus aufzuspüren. Befindet sich ein Gerät im Promiscuous-Modus, so fängt eine Netzkarte den gesamten Datenverkehr ab. Eine Netzschnittstelle, die in den Promiscuous-Modus versetzt wurde, ist ein deutlicher Hinweis auf das Vorhandesein eines Sniffers auf dem System. Das kostenlose Kommandozeilen-Tool *PromiscDetect* von NTSecurity bringt einen Mechanismus mit, der herausfinden kann, ob der Promiscuous-Modus aktiv ist.

5.5.2.8. Registry sichern

Obwohl die Werte in der Windows-Registrierungsdatenbank selbst nicht flüchtig sind, könnten sie weitere Ermittlungsschritte maßgeblich beeinflussen. Unter anderem sind folgende Registry-Werte zur Aufklärung entscheidend:

- ClearPageFileAtShutdown
- DisableLastAccess
- Autoruns

Windows-Registry besteht flüchtigen und nichtflüchtigen Daten. Zur Laufzeit des Betriebsystems wird die Registrierungsdatenbank im Arbeitsspeicher gehalten. Um (flüchtige) Daten der Registry zu sichern, bietet es sich an, ein Abbild des Hauptspeichers zu erstellen. Der nichtflüchtige Inhalt der Registry kann auch später aus einem forensischen Duplikat extrahiert werden. Wird kein Datenträgerabbild erstellt, so kann die Registry mit geeigneten Werkzeugen exportiert werden, um die nichtflüchtigen Komponenten der Windows-Registrierungsdatenbank zu sichern.

Weitere Informationen zu diesem Thema sind Kapitel 6 zu entnehmen.

5.5.2.9. Eventlogs sichern

Ein weiterer Ansatzpunkt bei den Ermittlungen sind die Windows-Eventlogs, in denen viele Programme eventuelle Fehler, Warn- oder Rückmeldungen hinterlassen. Es ist darauf zu achten, dass jede Tätigkeit auf dem

kompromitierten IT-System Einträge in den Protokollen hinterlässt. Es ist darüber hinaus darauf zu achten, dass, abhängig von der eingestellten Prokollierungsrate, es schnell passieren kann, dass ältere Meldungen überschrieben werden. Die Ereignisprotokolle des Computers können beispielsweise mit dem Dienstprogramm *PsLogList* eingesammelt werden. Detaillierte Erklärungen zu Windows-Logdateien folgen im Kapitel 6.

5.5.2.10. Abschließende Schritte

Nach dem Abschluss der Datensammlung wird die Referenzzeit nochmal erfasst. Dadurch lassen sich Spuren, die durch die Durchführung der Live Response von einem Ermittler unweigerlich zurückgelassen wurden, von den vom Angreifer hinterlassenen Indizien unterscheiden - eine anschließende Analyse hilft es herauszufinden, welche Dateien vom mutmaßlichen Täter gelesen, geändert oder ausgeführt wurden.

Anschließend wird das System hart heruntergefahren. Das bedeutet, dass ein physisches System vom Stromnetz genommen wird. Bei VMware ist die Sachlage etwas anders: je nach System kann das System entweder in den Suspend-Modus (Menüpunkt Anhalten) versetzt oder gänzlich heruntergefahren werden (Menüpunkt Ausschalten) - welche Vorgehensweise gewählt wird spielt keine Rolle. Der Hintergrund dafür ist, dass ein ordnungsgemäßes Herunterfahren den Zeitstempel des letzten Zugriffs vieler Dateien ändert, ganz egal, ob Dateien angelegt, geändert oder lediglich aufgerufen werden - das würde nicht reallozierten Platz gelöschter Dateien überschreiben und Wiederherstellung unmöglich machen. Angegriffene Systeme sind ferner oft mit Malware-Hintertüren infiziert, die permanent auf bestimmten Ports auf Verbindungsaufnahme des Hackers warten oder aber beim Herunterfahren eines Computersystems aktiv werden und weiterführende Änderungen am Host vornehmen bzw. Verschleierung seiner Spuren initiieren.

5.6. Datenerfassung unter SLES 11

Das folgende Kapitel beschreibt Methoden und Werkzeuge, die am Leibniz-Rechenzentrum eingesetzt werden, um an wichtige Informationen eines kompromittierten SuSE-Linux-Systems zu gelangen. Technische Grundlagen werden verkürzt dargestellt - deren Kenntnis erleichtert das Verständnis für die Arbeitsweise der vorgestellten Tools.

Ein Praxisunterschied zwischen den Windows- und Linux-Systemen stellt die Detektion der Schadprogramme dar - es kommt oft vor, dass ein Angreifer nach dem Ausführen das Binary löscht, Prozess selbst kann derweil weiterlaufen. Auch auf diese Problemstellung wird ebenfalls explizit eingegangen.

5.6.1. Live-Response-Toolkit

Kernphilosophie des SuSE Linux Live-Response-Toolkits ist der weitestgehende Verzicht auf die Verwendung der systemeigenen Befehle. Es kommt immer häufiger vor, dass Systemkomponente vom Angreifer ausgetauscht werden und dadurch gefälschte Ergebnisse liefern - dies kann Binärdateien (auch die Shell), Systembibliotheken oder den Kernel betreffen. Deswegen wird ganz bewusst nur vertrauenswürdigen, statisch gelinkten Werkzeugen gearbeitet [5.2]:

Tool	Beschreibung
bash	leistungsfähiger Kommandozeileninterpreter für Linux und Teil des GNU-Projekts.
cat	Das Kommando cat liest eine oder mehrere Dateien und schreibt das Ergebnis auf die Standardausgabe. Durch geschickte Parameterwahl lassen sich viele Systeminformationen auslesen, u.a. Kernelbuild, Anzahl der Prozessoren und deren Typ, RAM-Größe, usw.
df	hilfreiches Kommando zur Anzeige des freien Speicherplatz auf angeschlossenen Datenträgern.

fdisk	leistungsfähiges Befehlszeilentool, das die aktuelle Partitionierung der Festplatte(n) auf einem Computer anzeigt.
env	mit dem Befehl *env* lassen sich Umgebungsvariablen anzeigen.
who	hilfreiches Tool zur Überwachung von lokal & remote angemeldeten Benutzern.
ps	hilfreiches Kommando, um laufende Prozesse (auch in einer Baumstruktur) anzuzeigen.
arp	ein hilfreiches Tool zur Auflistung der Übersetzungstabellen für IP-Adressen, die von Address Resolution Protocol verwendet werden.
ifconfig	ein sehr wichtiges Programm, um aktive Netzschnittstellen sowie deren Statistik anzuzeigen. Wird auch benutzt, um neue Netzwerkinterfaces zu konfigurieren.
lsof	leistungsfähiges Befehl zur Auflistung aller geöffneter Dateien, Verzeichnisse, Unixsockets, IP-Sockets und Pipes.
ls	eines der wichtigsten Konsolenbefehle unter Linux - listet den Inhalt des aktuellen Verzeichnisses auf.
netstat	wichtiges Shell-Programm zum Abrufen der detailierten Protokollstatistiken und zur Anzeige aller aktiven TCP-, UDP- und IP-Verbindungen und Routingtabellen.
modinfo	ein Kommandozeilenbefehl zur Anzeige der Informationen über geladene Module/Kerneltreiber.
last	dieses kompakte Befehl zeigt den zuletzt eingeloggten Benutzer an.
file	hilfreiches Kommando, um Dateitypen zu identifizieren.
strace	wichtiger Shell-Befehl für die Analyse der laufenden Prozesse auf einem IT-System. strace zeichnet die Systemaufrufe eines laufenden Prozesses auf.
md5sum	ein Kommandozeilen-Programm zur Berechnung kryptographischer Hash-Funktion MD5.

Tabelle 5.2.: Bestandteile des SuSE Linux Enterprise Server Response Toolkits.

5.6.2. Live-Response

Die generelle Vorgehensweise ist vergleichbar zu Windows Server Systemen, es müssen gleiche Voraussetzungen erfüllt und relevante Fragestellungen beantwortet werden. Auch hier gilt: für virtuelle Systeme ist eine alternative Vorgehensweise denkbar. Sie wird im Abschnitt 5.6.2.9 vorgestellt.

Dabei gilt folgende Tätigkeitsreihenfolge:

1. vertrauenswürdige *Shell* aufrufen,
2. aktuelle Systemzeit und deren Differenz zu einer vertrauenswürdigen Referenzzeit erfassen,
3. Informationen über eingeloggte Benutzer sichern
4. Informationen über Netzverbindungen (offene Ports, aktive Verbindungen, gerade geschlossene Verbindungen, aktuelles Portmapping...) speichern
5. Sicherung der Informationen über laufende Prozesse (Mutterprozesse, Prozesseigentümer, Prozessorzeiten, Pfad, Aufrufparameter, ...)
6. Informationen über aktuellen Netzstatus bestimmen
7. Eventlogs sichern

8. aktuelle Systemzeit erfassen
9. Sicherung des /proc-Dateisystems,
10. Ergebnisse der Untersuchung sichern

Bei der Auswahl der Methodik und Hilfsmittel wurde darauf geachtet, dass sie in der Fachwelt allgemein anerkannt und beschrieben worden sind - eine wichtige Voraussetzung, damit die gewählte Vorgehensweise vor Gericht Bestand hat (vgl. 4.2).

Der Einsatz des Linux Incident-Response-Toolkits erfolgt naturgemäß in einer gewissen Reihenfolge. Dazu zählt das „Einlegen“ der ISO-Datei durch den zuständigen VMware-Administrator und das mounten der CD mit *mount /dev/cdrom0 /mnt/cdrom0* - der Pfad kann je nach System variieren.

Minimierung der Veränderung im Hostsystem spielt beim Live-Response von jeher eine bedeutende Rolle - entsprechend wird im nächsten Schritt eine vertrauenswürdige, statisch gelinkte Bash-Shell gestartet. Sie ist bereits vorkonfiguriert, d.h. sie nutzt lediglich mitgelieferte Binaries und minimiert Zugriffe auf Dateien des kompromittierten Systems.

5.6.2.1. Aktuelle Systemzeit erfassen

Sinnvollerweise wird zunächst die aktuelle Uhrzeit samt aktuellem Datum erfasst - hierfür wird lediglich der Befehl *date* aufgerufen. Abbildung 5.15 zeigt Beispiel für Ausführung der Kommandos.

```
ubuntu@ubuntu:~$ date
Thu Dec  1 21:28:22 UTC 2011
```

Abbildung 5.15.: Erfassung der aktuellen Systemzeit/des aktuellen Datums unter SLES 10

5.6.2.2. Inhalt des Kernel-Ringpuffers sichern

Während unter Windows Erstellung eines Arbeitsspeicherabbildes der nächste logische Schritt beim Live-Response wäre, kann unter Linux mit den gängigen Mitteln kein Hauptspeicherabbild angefertigt werden. Abschnitt 5.6.2.9 geht detailliert auf diese Problematik ein. Volatile Daten können daher lediglich strukturiert gesichert werden und zwar in der Reihenfolge ihrer Flüchtigkeit.

Der erste Schritt besteht insofern darin, den kompletten Ringpuffer des Unix-Kernels zu sichern - dort sind alle Meldungen zu finden, die noch nicht im Systemlog stehen. Mit dem Kommandozeilenprogramm dmesg können Kernel-Meldungen extrahiert werden. Der Befehl ist nützlich, um Kernel-Fehler oder Fehler beim Laden von Kernelmodulen ausfindig zu machen - immerhin agiert der Linux-Kernel als Schnittstelle zwischen dem Betriebssystem, den Anwendungen und der Hardware. Unter Umständen können Hinweise auf Kernel-basierte-Rootkits (LKM) gefunden werden. Durch den Umstand, dass sie in den Kernel geladen werden - und nicht im normalen Benutzermodus laufen - könnte das Laden eines Rootkit-Moduls mitprotokolliert werden. Nachfolgend wird ein Auszug aus der dmesg-Ausgabe, bei der eine Netzschnittstelle in den Promiscuous-Modus [5.6.2.6] versetzt wurde, gezeigt:

Es kann bereits an dieser Stelle sinnvoll sein, Details zur aktuellen Speichernutzung wegen ihrer hohen Flüchtigkeit zu erfassen - hierfür werden Daten im /proc-Dateisystem ausgewertet. Angaben zum aktuell belegten sowie freien physischen Speicher können aus */proc/meminfo* ausgelesen werden. Darüber hinaus können aus */proc/vmstat* statistische Informationen über den virtuellen Speicher angezeigt werden - damit lassen sich Zugriffe auf Swap- und Block-Geräte auswerten. Eine hohe Anzahl von Zugriffen auf den Swap-Bereich könnte ein Anzeichen für eine Ressourcenknappheit hindeuten, die durch den Einbruch verursacht wurde.

```
[  106.804527] NET: Registered protocol family 17
[ 7442.755589] end_request: I/O error, dev fd0, sector 0
[ 7442.775571] end_request: I/O error, dev fd0, sector 0
[ 7442.778768] program gparted is using a deprecated SCSI ioctl, please convert it to SG_IO
[195096.862985] device eth0 entered promiscuous mode
[195096.863702] audit(1323115018.986:2): dev=eth0 prom=256 old prom=0 auid=4294967295
```

Abbildung 5.16.: Über dmesg lässt sich Malware aufspüren.

Im Gegensatz zu *meminfo* und *vmstat* zeigt *loadavg* Entwicklung der Systemauslastung. Mit diesem Befehl kann die Anzahl der Prozesse, welche in der CPU-Wartheschlange auf Bearbeitung warten oder gerade verarbeitet werden, in den letzten 60 Sekunden, der letzten 5 sowie der letzten 15 Minuten angezeigt werden. Der gemittelte Wert wird vom Kernel ermittelt. Generell gilt: je höher der Wert ist, desto größer ist die Systemauslastung. Beispielsweise verursachen Sniffer, die den gesamten Netzverkehr protokollieren, eine hohe Systemlast - weitere Anhaltspunkte könnte in diesem Fall Analyse der laufenden Prozesse liefern.

5.6.2.3. Informationen über eingeloggte Benutzer sichern

Bei jeder forensischen Analyse ist es wichtig, eine Liste der aktuell lokal und - insbesondere - remote eingeloggten Benutzer zu sichern. Hierzu gehören nicht nur die Auflistungen aller aktiven Benutzer, sondern auch der Inhalt der Datei */etc/passwd* - sie enthält alle Informationen über Benutzerkonten, insbesondere den Benutzernamen, Heim-Verzeichnis und das sog. Login-Kommando [4] Unter anderem gilt es zu klären, ob Verstöße gegen Sicherheitsrichtlinien vorliegen [CER]:

- Sind oder waren Benutzer zu ungewöhnlicher Zeit angemeldet?
- Gab es Logins von ungewöhnlichen Systemen?
- Gibt es neue Benutzer, die so nicht auf das System gehören?
- Haben bekannte oder neue Nutzer erweiterte Privilegien?
- User-ID '0' sollte nur für root vergeben sein.

Um diese Informationen zu ermitteln, wird auf folgende Dateien zugegriffen: */etc/passwd*, *utmp* sowie *wtmp* - dabei werden folgende Befehle verwendet: *cat /etc/passwd*, *who* sowie *last*.Die MAC-Zeiten der genannten Dateien werden dabei modifiziert. Weiterführende Informationen können dem Unterkapitel 5.6.2.3 entnommen werden.

5.6.2.4. Informationen über Netzverbindungen speichern

An dieser Stelle ist es sinnvoll, Informationen über aktive Netzverbindungen sicherzustellen - sofern nach dem Bekanntwerden eines Security Incidents die Verbindungen noch nicht getrennt wurden. Gelegentlich lässt sich Malware anhand der aus- bzw. eingehenden Verbindungen identifizieren, auch eine Identifikation des Angreifers ist nicht vollständig auszuschließen. Es sollte damit gerechnet werden, dass selbst wenn das betroffene System weiterhin ans Netz angebunden ist, bestehende Netzverbindungen zu jedem Zeitpunkt geschlossen werden können und sich nicht mehr nachweisen lassen. Folgende Fragen sind zu klären [CER]:

- Haben ungewöhnliche Prozesse eine Netzwerkverbindung oder nehmen sogar selber auf einem Port Verbindungen an?
- Läuft ein Dienst, der nicht laufen sollte bzw. sind neue Dienste dazugekommen?
- Wird von einem bekannten Service ein falscher Port gebunden?

Auch an dieser Stelle wird auf die Funktionalität der normalen Linux-Befehle zurückgegriffen: *netstat* sowie *lsof*. Zu beachten ist, dass im Gegensatz zu Windows-basierten Systemen [5.5.2.6], in diesem Schritt neben

[4] Der Befehl, der nach dem Einloggen ausgeführt wird.

aktiven Netzverbindungen auch gleichzeitig aktuelles Portmapping ermittelt wird - dadurch soll geklärt werden, ob der Hacker eine Hintertür installiert und dabei einen entsprechenden Port geöffnet hat. Bei der Suche nach unbekannten Portnummer leisten einschlägige Suchmaschinen gute Dienste.

netstat ist ein Kommandozeilenprogramm, mit dem sich alle aktiven Netzverbindungen und detaillierte Protokollstatistiken abrufen lassen. Das Tool lässt sich über einige Parameter steuern, die beim Aufruf eingegeben werden können. Typischerweise werden folgende drei Parameter verwendet: *-a -p -n*. *-a* zeigt eine Liste aller aktiven Verbindungen an, mit dem Argument *-p* erhält man zugehörige Programmnamen. Das Programmoutput wird mit dem Schalter *-n* beschleunigt, da Adressen nicht aufgelöst werden. Zusätzlich lässt sich noch der Schalter *–inet* bzw. *–inet6* verwenden - dadurch lässt die die Ausgabe auf IP-Sockets der IP-Version 4 bzw. 6 eingrenzen. Die Ausgabe sieht dann folgendermaßen aus:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address      Foreign Address     State       PID/Program name
tcp        0      0 127.0.0.1:631      0.0.0.0:*           LISTEN      17707/cupsd
tcp6       0      0 :::22              :::*                LISTEN      17567/sshd
udp        0      0 0.0.0.0:68         0.0.0.0:*                       18261/dhclient
udp        0      0 0.0.0.0:48474      0.0.0.0:*                       17590/avahi-daemon:
udp        0      0 0.0.0.0:5353       0.0.0.0:*                       17590/avahi-daemon:
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags    Type    State      I-Node  PID/Program name   Path
unix  2      [ ACC ]  STREAM  LISTENING  67920   18745/gam_server   @/tmp/fam-ubuntu-
unix  2      [ ACC ]  STREAM  LISTENING  64196   18668/dbus-daemon  @/tmp/dbus-tBZHkUwM65
unix  2      [ ]      DGRAM              14810   1/init             @/com/ubuntu/upstart
unix  12     [ ]      DGRAM              60735   17371/syslogd      /dev/log
unix  2      [ ACC ]  STREAM  LISTENING  60307   17174/acpid        /var/run/acpid.socket
(...)
```

Listing 5.2: Ausgabe des „netstat -anp“ Befehls

Das Programmoutput ließe sich weiter eingrenzen. Zum Beispiel listet folgender Befehl *netstat -apn |grep ":6667“* | lediglich bestehende Verbindungen für Port 6667 (Internet Relay Chat) an - dieser Port wurde in der Vergangenheit oft für Hintertüren verwendet, so wie beim trojanischen Pferd Maxload [Sym].

lsof (list of open files) ist ein hilfreiches Shellprogramm, u.a. auch zum Aufspüren geöffneter Netzverbindungen. Folgende Parameter werden verwendet: *-P* erzwingt Anzeige von Portnummern statt Servicenamen, *-n* verhindert das Auflösen von Hostnamen, mit dem optionalen Argument *-i* erhält man detaillierte Auflistung aller IP-Sockets. Schließlich legt man mit dem Parameter *-V* fest, dass lsof eine Meldung ins Log schreibt, selbst dann, wenn keine Verbindungen erkannt wurden [5.17].

```
ubuntu@ubuntu:~$ lsof -P -n -i -V
lsof: no Internet files located
ubuntu@ubuntu:~$ lsof -P -n -i
ubuntu@ubuntu:~$
```

Abbildung 5.17.: Beispielausgabe des CLI-Tools lsof mit und ohne des -V Schalters

Während netstat Verbindungen in bündiger Form auflistet und eine schnelle Analyse ermöglicht, kann mit lsof eine wesentlich detailliertere Überprüfung vorgenommen werden.

Hilfreich ist auch ein externer Scan des Systems von einem “benachbarten“ Host über das lokale Intranet - der bekannteste Portscanner dürfte nmap von Gordon Lyon sein [Lyo]. Damit kann man den betroffenen Host von außen inspizieren und unter Umständen einer Malware auf die Schliche kommen. Existieren Diskrepanzen zu den Ergebnissen von lsof bzw. netstat, so ist das meist ein Indiz für das Vorhandesein eines Rootkits und dessen Netzwerkaktivitäten zu werten.

Folgende Reihenfolge bietet sích an:

- *-sS IP-Adresse*: es wird ein TCP SYN Scan [2.2.2] der standardisierten (well known) Ports ausgeführt.
- *-sS -p 1-65535 IP-Adresse*: es wird ein TCP SYN Scan aller TCP-Ports ausgeführt.
- *-sU IP-Adresse*: es wird eine Suche nach offenen UDP-Ports [2.2.2] ausgeführt (u.U. langsam).

Zu beachten ist: Eine Spezialform der Hintertüren, sog. Kernel-Backdoor, versucht erst dann eine Verbindung zu einem Server aufzubauen, wenn ein speziell präperiertes Packet auf einem vordefinierten Port eintrifft. Das heißt in der Praxis, dass Rootkits die Kernel-Arbeitsweise zunutzemachen: das Backdoor läuft nicht permanent, sondern wird erst nach Erhalt eines spezifischen Datenpakets ausgeführt. Dieser Tarnmechanismus ist auch unter der Bezeichnung Portknocking bekannt. Damit sind solche Hintertüren nur schwer aufzuspüren [EYV].

5.6.2.5. Sicherung der Informationen über laufende Prozesse

Die Sicherung der Informationen über alle laufenden oder einzelne Prozesse gehört zu den zentralen Anforderungen beim Live Response. Einen Überblick über Prozesse im System gibt der Aufruf des Shell-Tools *ps* mit den Parametern „-efl“: Der Schalter *-e* bewirkt dabei Anzeige aller Prozesse, *-f* erzwingt sog. volles Format und durch die Verwendung der Option *-l* werden alle Prozessattribute aufgelistet. Üblicherweise erhält man folgende Informationen:

- absoluter Pfad zur ausführbaren Datei,
- Aufrufparameter (sofern verwendet),
- Prozessstartzeit,
- Prozesslaufzeit,
- Prozesseigentümer,
- Prozessorzeiten,
- Prozess-ID,
- Mutter-Prozess-ID.

Die Auswertung der Ausgabe kann dabei helfen bösartige Prozesse aufzuspüren. Falls ein Prozess unter einer falschen Benutzerkennung läuft, ist es ein Hinweis auf unerwünschte Aktivitäten. Das gleiche Prinzip greift auch, falls die Prozess-ID eines Dienstes sehr hoch ist, obwohl er während oder kurz nach dem Bootvorgang gestartet werden sollte. Auch der vollständige Pfad zur ausführbaren Datei und verwendete Aufrufoptionen sind vom großen Wert für die Ermittler und bilden einen weiteren Ausgangspunkt für weitere Maßnahmen. Eine umfassende Kenntnis der laufenden Prozesse ist als Grundlage unabdingbar, um unerwünschte Prozesse entdecken zu können. Erst wenn laufende Prozesse detailliert erfasst sind, kann über konkrete Gegenmaßnahmen nachgedacht werden. Bei der Entdeckung eines unbekannten Prozesses mittels *ps -efl* ließen sich entsprechende Dateien sicherstellen, selbst dann, wenn ein Programm nach dem Start gelöscht wurde - man kann das zugehörige Binary über das Prozess-ID im proc-Dateisystem unter */proc/PID/exe* finden und, unter forensisch korrekten Gesichtspunkten, für die spätere Analyse sichern. Der Live-Response-Skript kann automatisch alle Prozesse aus dem /proc-Dateisystem auslesen und entsprechend sichern - dieser Vorgang nimmt eine gewisse Zeitspanne in Anspruch. Die nachfolgende Listendarstellung gibt einen kleinen Einblick in die Vorgehensweise - die Ausgabe kann (und muss) natürlich in eine Datei umgeleitet werden:

```
cat /proc/$PID/cmdline
cat /proc/$PID/environ
cat /proc/$PID/maps
cat /proc/$PID/stat
cat /proc/$PID/statm
cat /proc/$PID/status
ls -ld /proc/$PID/root
ls -ld /proc/$PID/cwd
ls -ld /proc/$PID/exe
ls -lrta /proc/$PID/fd/
```

Listing 5.3: Beispielabfolge von Befehlen zur Sicherung des Prozessspeichers

Sicherlich darf man Malwareautoren nicht unterschätzen - sie variieren die Methoden zur Tarnung eines Schadcodes, kombinieren sie und hebeln klassische Systemfunktionen teilweise aus. Zudem verwenden sie verschie-

dene Ebenen des Betriebssystems, um Schadsoftware unbemerkt einzunisten. Korreliert man erhaltene Informationen mit der Auswertung des aktuellen Portmappings (vorgestellt auf den nachfolgenden Seiten) und den Ergebnissen des Netzmonitorings [5.7], könnte der Grund für ungewöhnliches Systemverhalten identifiziert werden.

5.6.2.6. Informationen über aktuellen Netzstatus bestimmen

Installierter Schadcode startet häufig einen Sniffer, der den Datenverkehr ausspäht, alle gesendeten und empfangenen Pakete mitprotokolliert, unter Umständen auswertet und unbemerkt per Internet weiterleitet. Dafür versetzt Malware in der Regel einen Netzwerkadapter in den sog. Promiscuous-Modus [5.5.2.7]. Eine sehr einfache Überprüfung lässt sich mit dem Aufruf *netstat -i* durchführen. Das Programm scannt alle aktiven Netzschnittstellen und gibt das Ergebnis auf dem Bildschirm aus. Ist in der letzten Spalte der Flag **P** gesetzt (so wie in der Abbildung 5.18 zu sehen), so ist es ein deutliches Indiz auf das Vorhandensein eines Sniffers auf dem System. Die Verwendung von Switches am Leibniz-Rechenzentrum führt allerdings dazu, dass im Regelfall ein System nur eigenen Datenverkehr sowie Broadcasts anderer Systeme empfangen kann.

```
ubuntu@ubuntu:/$ netstat -ai
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500 0      2026      0      0 0           67      0      0      0 BMRU
lo        16436 0      1990      0      0 0         1990      0      0      0 LRU
ubuntu@ubuntu:/$ sudo ifconfig eth0 promisc
ubuntu@ubuntu:/$ netstat -ai
Kernel Interface table
Iface   MTU Met   RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0       1500 0      2027      0      0 0           68      0      0      0 BMPRL
lo        16436 0      1990      0      0 0         1990      0      0      0 LRU
```

Abbildung 5.18.: Netstat kann Schnittstellen im Promiscuous-Modus aufspüren

Anschließend kann selbstverständlich die detaillierte Netzwerkkonfiguration abgefragt werden. Der Befehl dafür heißt *netstat -aei* - die Ausgabe entspricht dem Format von *ifconfig -a*.

Für Datentransport zwischen verschiedenen Subnetzen werden Routingtabellen verwendet. Hat ein Angreifer Root-Rechte erlangt, hat er die Möglichkeit bestimmte Teile der Kommunikation zu steuern - mit anderen Worten: Routing und Identifizierung. Das kann dazu führen, dass Routingtabellen modifiziert werden. Durch einen gefälschten Routingtabelleneintrag wird der Datenfluss beispielsweise zu einem alternativen, speziell präparierten Endpoint umgeleiten.

Aktuelle Routingtabellen können mit dem Befehl *netstat -rn* abgerufen werden, wobei der Paramter *-r* für Routing steht. Der Schalter *-n* unterdrückt, wie gewohnt, DNS-Lookup.

Neben dem Ändern von Routingtabellen gibt es eine weitere Methode, um Verbindungen in Netzen umzuleiten und zu belauschen: Modifikation der ARP-Tabellen, sog. ARP-Spoofing. Dabei wird der ARP-Cache mit manipulierten Datenpaketen so modifiziert, sodass Verbindungen zwischen Server und Client über das kompromittierte System, ähnlich wie ein Proxy-Server, geleitet werden. Bei dieser Form des Man-in-the-Middle-Angriffs lässt sich der Datenverkehrt mitlauschen - sofern keine Verschlüsselung zum Einsatz kommt. Das am LRZ eingesetzte Intrusion-Detection-System Snort kann ARP-Spoofing durch Inkonsistenzen bei MAC-Adressen erkennen, wenn auch erst nach dem Beenden des Angriffs. Der Vollständigkeitshalber sollten aktuelle ARP-Tabellen für die sichere Beweisführung trotzdem gesichert werden. Die Einträge können mit dem Befehl *arp -nv* untersucht werden. Wie gewoht, wird die DNS-Suche mit dem Schalter *-n* unterdrückt, um Informationen numerisch darzustellen und die Anfragebearbeitung zu beschleunigen. Sind im ARP-Cache mehrere Einträge mit identischen MAC-Adressen vorhanden, ist das ein Indiz für umgeleitete Verbindungen.

```
ubuntu@ubuntu:/media$ netstat -aei
Kernel Interface table
eth0      Link encap:Ethernet  HWaddr 00:50:56:8f:01:c8
          inet addr:192.168.247.10  Bcast:192.168.247.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe8f:1c8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1856 errors:0 dropped:0 overruns:0 frame:0
          TX packets:31 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:360236 (351.7 KB)  TX bytes:4771 (4.6 KB)
          Base address:0x2000 Memory:d8920000-d8940000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1982 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1982 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:104596 (102.1 KB)  TX bytes:104596 (102.1 KB)
```

Abbildung 5.19.: Netstat überprüft Schnittstellenkonfiguration und stellt sie übersichtlich dar.

5.6.2.7. Eventlogs und Konfigurationdateien sichern

Abhängig von der Systemkonfiguration werden Logdateien am LRZ zum Teil lokal, zum Teil auf einem der zentralen Logserver gesichert. Sofern sie nicht gelöscht wurden, besteht bei Protokolldateien vom angegriffenen System immer der Verdacht, dass sie kompromittiert sein könnten. Es empfiehlt sich trotzdem, wo vorhanden und sinnvoll, Daten über Systemnutzung zu sicher - sie könnten zur Aufklärungen eines sicherheitsrelevanten Vorfalls beitragen sowie für Beweisführung in einem Rechtsfall unabdingbar sein. Unter anderem versucht man anhand von Logging-Informationen nachzuvollziehen, welcher Angriff durchgeführt wurde und ob eine Schwachstelle den Angriff begünstigt hat.

Unabhängig davon existieren zwei Möglichkeiten zur Sicherung von Logdateien:

- Sicherung und Analyse während der Live-Response,
- Wiederherstellung aus einem forensischen Duplikat.

Jeglicher Zugriff auf Dateien hat natürlich eine Änderung ihrer MAC-Zeiten zur Folge, daher gilt es immer Zeitdauer für die Duplikation mit der Beweishärte abzuwägen.

Unter SLES existiert eine Myriade von Logdateien. Typischerweise werden sie im Verzeichnis */var/log/* gespeichert. Eine kleine Auswahl bietet folgende Auflistung, wobei im Endeffekt das Loggingverhalten von der Konfiguration des syslog-Daemons abhängig ist:

- */var/log/messages*: protokolliert alle Systemereignisse
- */var/log/mysqld.log*: MySQL Log
- */var/log/apache2/*: enthält diverse Logs, die vom Apachmodul erstellt werden.

Unabhängig davon sollten diverse Systeminformationen auf unterschiedlichen Ebenen sichergestellt werden. Standardmäßig sollten folgende Informationen für die spätere Auswertung sichergestellt werden:

- Hostname sowie Betriebszeit,
- Kernelversion - beantwortet die Frage, ob das System von einer Schwachstelle betroffen sein kann,
- Umgebungsvariablen für Programme,
- Parameter, mit denen der Kernel gestartet wurde,
- Liste aller Module an, die im Kernel geladen wurden,

- Kernel-Parameter,
- Symboltabelle des Kernels,
- diverse Informationen zur Hardware.

5.6.2.8. Abschließende Schritte

Nachdem die Datensammlung abgeschlossen ist, müssen gleiche Schritte wie bei Windows-basierten Systemen getroffen werden [5.5.2.10]. Das bedeutet in erster Linie nochmalige Erfassung der Referenzzeit und das harte Herunterfahren des kompromittierten Systems. Auch hier lässt sich die alternative Vorgehensweise bei virtuellen Maschinen anwenden.

5.6.2.9. Arbeitsspeicherabbild erstellen

Während unter Windows Erstellung eines Arbeitsspeicherabbildes eines der ersten Schritte beim Live-Response ist, gestaltet sich das bei Linux etwas problematisch. Ein Speicherabbild ließ sich beim Kernel 2.4 noch unter Verwendung des Werkzeugs *dd* und dem Aufruf von *dd if=/dev/kmem of=/root/kmem* bzw. *dd if=/dev/mem of=/root/mem*[5]) ([eHo]) erstellen. Unter Linux Kernel 2.6 wurde *kmem* entfernt, und der Zugriff auf *mem* stark beschränkt, so dass nur noch das erste Megabyte des Speichers extrahiert werden konnte. Mit der fortschreitenden Entwicklung des Kernels wurde der Umfang des Zugriffs auf das RAM über virtuelle Geräte immer weiter eingegrenzt - in den neuesten Iterationen des Kernels ist *dev/mem* in der Standardkonfiguration ebenfalls nicht mehr vorhanden.[Wikc]). Nur mit Einsatz von kommerzieller Software (z.B. Second Sight), die ein Kernel-Gerätetreiber installiert, um aus dem Kernelmode ein Abbild des physikalischen Speichers zu erstellen, ist es zum Zeitpunkt der Erstellung dieses Leitfadens möglich ein RAM-Dump zu generieren.

Nichtsdestotrotz ist es zumindest bei virtuellen Maschinen möglich, ein Arbeitsspeicherabbild zu erstellen. Auch hier wird auf die VMware-Funktion „Snapshot erstellen...“ (engl. create snapshot) [5.5] zurückgegriffen - der Inhalt des Arbeitsspeichers landet in der Datei mit der Endung .vmem. Die VMEM-Datei kann mit den gängigen Memory-Analyse-Programmen (z.B. Volatility Framework in der aktuellsten Version oder Volatilitux (nur x86-Architektur), [Wika]) analysiert werden. Das Vorgehen zur Erstellung eines Speicherabbilds wurde bereits im Abschnitt 5.5.2.2 kurz angeschnitten. Detaillierte Informationen zur Arbeitsspeicher-Analyse sind im Abschnitt 6.3.2.1 zu finden.

Die Nutzung der Funktion „Snapshot“ eröffnet neue Möglichkeiten und eine engere Verflechtung mit dem SIR-Prozess. Es ist sogar denkbar, auf den Einsatz des LR-Toolkits gänzlich zu verzichten und lediglich ein Arbeitsspeicher-Abbild zu generieren - immerhin wird bei einem Systemeinbruch meistens eine Trennung aller Netzverbindungen initiiert. Voraussetzung ist natürlich ein stabiles, zuverlässiges Framework für die Analyse der Linux-Hauptspeicherabbilder (siehe Abschnitt 6.3.2.1). Wird ein System in den Suspend-Modus versetzt, lassen sich durch Analyse der VMEM-Dateien zur Laufzeit aktive Prozesse, Threads und Netzverbindungen analysieren. Des Weiteren können bereits beendete Prozesse und Netzverbindungen identifiziert werden - jedenfalls falls entsprechende Speicherbereiche noch nicht mit neuen Daten gefüllt wurden. Denkbar ist auch der Einsatz einer um einige Funktionen reduzierten Version des Live-Response-Toolkits - nur die Informationen, die nicht durch ein Hauptspeicherabbild abgedeckt werden, werden gesammelt - u.a. kann auf das langwierige und fehleranfällige Auslesen des Prozess-Speichers verzichtet werden.

5.7. Datenerfassung im Netz

Das Thema Netzwerk-Forensik ist ein relativ junges Fachgebiet der Computer-Forensik. Die grundsätzliche Problematik beim Erlangen von digitalen Netzwerkspuren ist die komplexe Netzinfrastruktur am LRZ sowie Limitierungen bzw. Beschränkungen diverser Monitoringtechnologien. Nicht immer können verdächtige Anomalien einfach erkannt und aufgezeichnet werden. Weiteres Problem: es müssen oft viele GigaByte von Netzwerkdaten analysiert werden, wobei gelegentlich nicht genau ersichtlich ist, wonach man suchen muss.

[5] Virtuelle Komponente, die den Arbeitsspeicher des Rechners abbilden.

Wie in Kapitel 3 beschrieben werden am LRZ verschiedene professionelle Werkzeuge für Netz-Monitoring eingesetzt. In der Regel werden einzelne Werkzeuge kombiniert, um unterschiedliche digitale Dateninformationen zu erlangen. Die Datenerfassung im Netz besteht überwiegend aus der Extraktion von Daten aus relevanten Logeinträgen - sie kann in den Abschnitt der Datensammlung des forensischen Prozesses eingeordnet werden - und Echtzeitüberwachung der betroffenen Systeme sowie ihrer mittelbaren Umgebung - ein Teil des LRZ-SIR-Prozesses.

Die Notwendigkeit des Netz-Monitorings ist auch in der Tatsache begründet, dass nach einem erfolgreichen Einbruch vom Angreifer Rootkits installiert sowie ein Hintertür eingerichtet werden. Moderne Rootkits greifen direkt auf den im Hauptspeicher befindlichen Kernel und schaffen es so wirkungsvoll Hinweise auf Kompromittierung des Systems zu verwischen: Dateien und Prozesse werden versteckt, Systemprogramme durch trojanisierte Tools ersetzt, Logdateien werden bereinigt. Abhängig von der Komplexität des Rootkits kann Live-Response unter Umständen keine Ergebnisse liefern. Genau an dieser Stelle setzt Netz-Monitoring an.

Wesentliches Ergebnis der Netzüberwachung in Netzen ist die Schaffung der Klarheit, ob weitere Systeme in der Umgebung kompromittiert wurden. Auch kann anhand des Monitorings von Kommunikationsverhaltens die Spurensuche auf bestimmte (Teil-)Bereiche der Forensik eingegrenzt werden.

Netz-Infrastruktur am LRZ ist hoch dynamisch und wird laufend überwacht. Daten werden i.d.R. im Vorfeld, d.h. proaktiv, erhoben und für einen Zeitraum von 7 Tagen aufbewahrt. Im Bereich des Netz-Monitorings spielen folgende Aspekte eine zentrale Rolle:

- Mitschnitt des kompletten Datenverkehrs (für das LRZ ist dieses Vorgehen nicht profitabel),
- Ereignisüberwachung (Event Monitoring, es wird jede bekannte Art von Anomalien festgehalten),
- Netflow-Analyse (Trap-and-Trace Monitoring, es werden nur die sog. Verkehrsdaten festgehalten).

Auf die letzten zwei Bereiche wird in den folgenden Unterabschnitten eingegangen. Grundlegende Informationen zu den verwendeten Softwarelösungen wurden bereits im Kapitel 3 behandelt.

5.7.1. Ereignisüberwachung

Wie in Abschnitt 3.1.1 beschrieben wird am LRZ zur Ereignisüberwachung in Netzen das Werkzeug Snort eingesetzt. Snort ist ein netzwerkbasiertes Intrusion Detection System, das u. a. den Datenverkehr eines Netzsegments überwacht und analysiert, um daraus Muster abzuleiten, die auf einen Angriffsversuch im Netz hindeuten könnten. Folgende Sicherheitsverstöße lassen sich identifizieren (Erkennungsmuster müssten zwingend bekannt sein):

- Pufferüberlauf,
- Portscans,
- DDoS-Angriffe,
- CGI-Angriffe,
- ARP-Spoofing/Cache-Poisoning,
- ...

Die Praxis zeigt, dass viele Einbruchsversuche oder gar erfolgreiche Einbrüche zuverlässig erkannt werden - vorausgesetzt es stehen entsprechende Signaturen zur Verfügung.

5.7.2. Analyse von NetFlow-Daten

Eine detaillierte Betrachtung des Datenverkehrs ist mittels Netflow-Analyse möglich. Für die Überprüfung und Rekonstruktion der Flussdaten kommt am Leibniz-Rechenzentrum die Toolsammlung nfdump samt Web-Frontend NfSen für eine grafische Übersicht zum Einsatz [3.1.4]. Mit NfSen ist es ganz einfach möglich, eine übermäßige Nutzung von Netzwerkbandbreiten und einen unvorhergesehenen Datenverkehr zu erkennen und

zu isolieren. Für eine einfachere Überprüfung der Flussdaten wird der Datenverkehr nach bestimmten Kriterien (z.B. Systemen, Schnittstellen, Ports, Protokollen, Diensttyp, Quell-IPs, Ziel-IPs, usw.) gefiltert sowie gruppiert und ermöglicht dem CSIRT-Team einen umfassenden Einblick in Nutzungsdaten und Kommunikationsverhalten. NfSen wird am LRZ gemeinhin verwendet für:

- Identifizierung von Angriffen oder von ungewöhnlichen Altivitäten im Netz, ohne dass eine Signatur- oder Muster-Liste benötigt wird,
- Überwachung des Verkehrs, der von bestimmten Ports, Quell-IPs, Ziel-IPs, Schnittstellen und sogar von Protokollen eingeht,
- Durch Echtzeitüberwachung der NetFlow-Daten entstehen detaillierten Informationen, die zur Eingrenzung und Diagnose von kompromittierten Systemen verwendet werden können.

Dank der permanenten Kontrolle des ein- und ausgehenden Datenverkehrs kann die forensische Analyse sehr weit ins Detail gehen: es lassen sich benutzerdefinierte Filter anlegen, mit deren Hilfe das LRZ-CSIRT-Team problematische Systeme schnell identifizieren und fortlaufend im Auge behalten kann.

```
tcp and ( src ip 192.168.247.10 or dst ip 60.190.222.139)
tcp and ( net 192.168/16 and src port 65520 and dst ip 60.190.222.139 )
```

Listing 5.4: Beispiele für benutzerdefinierte NfSen-Filter

5.7.3. Analyse von Accounting-Daten

Erkennung von Botnetzkommunikation und die Identifikation von Spamverteilern erfolgt mithilfe eines ratenbasierten Monitoringmechanismus (vgl. Abschnitt 3.1.3). Gegenüber Snort und NfSen bietet er eine deutlich bessere Flexibilität bei der Ermittlung der IP-Adressen, die ungewöhnlich viel Datenverkehr im Netz verursachen. Ein weiterer wesentlicher Vorteil ist die Erkennung von 0-Day-Attacken durch die Auswertung der ein- bzw. ausgehenden Datenaufkommen der Systeme.

Geplant ist, dass Daten aus verschiedenen Quellen in OSSIM aggregiert werden können - im Moment funktioniert dies nur bei den von Snort gemeldeten Eereignissen: alle Snort-Events werden zentral gesammelt, korreliert, ausgewertet und entsprechend zuständige Administratoren automatisiert informiert. Dies soll einen schnellen Drilldown in den Datenverkehr für bestimmte Netzelemente ermöglichen. Die Nutzung eines zentralen Auswertungsystems würde dann zu effizienten und schnellen Prozess-Schritten führen: Möchte man zum Beispiel wissen, wie welche Vorkommnisse zusammenhängen, liessen sich verschiedene Sicherheitsmeldungen miteinander verknüpfen und durch Einsatz diverser Filter weiter verfeinern. So wäre es bei Verdacht auf Botnetzkommunikation beispielsweise möglich, eine Ziel-IP-Adresse zu definieren und damit nur die Verbindungen von den lokalen Systemen zu dieser Adresse anzuzeigen. Gleiches Vorgehen würde auch für Angriffe über eine Schnittstelle, beispielsweise für den SSH-Dienst, funktionieren. Das Endergebniss könnte in einer Datei exportiert und auf dem zentralen Forensik-Server abgelegt werden.

Generell gilt: mit dem Netz-Monitoring kann eine deutliche Reduzierung der Ermittlungsaufwände und -kosten erreichen. Entscheidend ist die ganzheitliche Berücksichtigung der organisatorischen und prozessuellen Abhängigkeiten. Das wird deutlich, wenn das CSIRT-Team die Ergebnisse der von System- bzw. Netz-Administratoren durchgeführten Untersuchungen mit den Ereignissen im lokalen LRZ-Netz korrelieren. Durch die Verbindung alle Ergebnisse lässt sich (evtl. mit der Hinzunahme der Post-Mortem-Analyse) ein zeitlicher Verlauf des Angriffs rekonstruieren. Gleichzeitig führen die Erkenntnisse zur einer Qualitätssteigerung bei den Recovery-Maßnahmen.

5.8. Forensische Duplikation

Das Duplizieren der Festplatten-Daten ist, neben dem Arbeitsspeicherabbild, entscheidend für die nachfolgenden Abschnitte des forensischen Ermittlungsprozesses. Forensische Duplikation stellt Unversehrtheit der digitalen Beweise sicher und erlaubt eine tiefgehende Analyse der Vorgänge - manchmal auch entscheidend

für den Ausgang einer Ermittlung. Dieser Abschnitt definiert den Begriff „forensische Duplikation“, und beschreibt Methoden und Tools, die einem Ermittler helfen, nichtflüchtige Daten auf Datenträgern eines Computersystems zu erfassen.

5.8.1. Was ist forensische Duplikation

Bevor das Vorgehensweise zur Datenträger-Duplikation erläutert und einige Basiswerkzeuge vorgestellt werden, soll zunächst der Begriff der „forensische Duplikation“ formal eingeführt werden.

> *Ein forensisches Duplikat ist ein Datenträgerabbild, das bitweise als eine 1:1-Kopie erstellt wurde. Hierbei werden die nichtflüchtigen Daten auf Datenträgern eines Computersystems erfasst.*

Technisch resultiert eine 40 GB große Festplatte in einem 40 GB großen Image - immerhin wird jedes Bit des Untersuchungsmediums gesichert. Zur allgemein akzeptierten Vorgehensweise gehört, dass nur dann Veränderungen am Image während des Kopiervorgangs vorgenommen werden, wenn beim Lesen Fehler aufgetreten sind - nach mehrfachen Leseversuchen wird der fehlerhafte Sektor markiert und mit einem Platzhalter versehen. Nach dem Imaging-Vorgang muss die Kopie verifiziert und gegen Änderungen geschützt werden. Das erfolgt in der Regel durch den Einsatz kryptografische Verfahren (Checksummen oder Hash-Algorithmen), die die Integrität des Duplikats gewährleisten sollen.

5.8.2. Ist forensische Duplikation notwendig?

Bevor die Entscheidung über Durchführung einer forensischen Duplikation getroffen wird, sollte eine grundlegende Analyse durch den Security Incident Coordinator (SIC) und CSIRT-Forensiker durchgeführt werden. Die Analyse dient der Prüfung, ob die Gewinnung eines Datenträgerabbildes notwendig bzw. vertretbar ist. Die Erfahrung zeigt, dass, obwohl eine forensische Duplikation grundsätzlich ratsam ist, ein längerfristiger Ausfall der kritischen Systeme durch die LRZ-Betreiber kaum toleriert werden kann - die Faktoren Datenvolumen und Zeit stehen hier in einem direkten Verhältnis zueinander. Die Zeitdauer für die Erstellung einer 1:1-Kopie steigt mindestens proportional zur Datenmenge. Dieses Bewusstsein muss bei den Mitgliedert des LRZ-CSIRT vorhanden sein. Weiterhin ist bei den Überlegungen entsprechend zu berücksichtigen, dass mit steigender Datenmenge auch der Analyseaufwand steigt - die Ermittlertätigkeiten könnten allerdings parallel abgearbeitet werden, aufgrund der benötigten Abstimmung müsste dies durch den SIC koordiniert werden. Die verursachten Aufwände sind im Verhältnis zu möglichen Erkenntnissen zu sehen. Daher gilt es zunächst, folgende Fragen zu beantworten (vgl. auch Abbildung 5.20):

- Handelt es sich um einen bekannten Sicherheitsvorfall? [FA]
- Handelt es sich um ein „kritisches“ System? [FB]
- Kann ein längerer Ausfall der Komponente toleriert werden? [FC]
- Entsteht durch die Kompromittierung und den dadurch bedingten Systemausfall ein (hoher) finanzieller Schaden? [FD]
- Hat Live-Response ausreichend Anhaltspunkte geliefert, um Methode oder Schwachstelle zu identifizieren, die zum Systemeinbruch geführt haben? [FE]
- Muss ein Datenträger analysiert werden? [FG]
- Kann ein Ermittlungsverfahren eingeleitet werden? [FH]

Grundsätzlich gilt: kann eine der folgenden Fragen, nämlich [FC], [FD], [FG] oder [FH], „bejaht“ werden, sollte forensische Duplikation durchgeführt werden.

5.8.3. Qualifiziertes forensisches Duplikat

Wenn es darum geht, ein forensisches Duplikat eines physischen oder virtuellen Datenträgers anzufertigen, existiert eine Reihe von Möglichkeiten.

- **AIR** (Automated Image and Restore) [Abb. 5.21] - AIR ist ein Softwareprodukt von Steve Gibson und Nanni Bassetti und bietet eine anwenderfreundliche Oberfläche. Genau genommen handelt es sich dabei um ein Frontend für die Linux-Tools dd sowie dcfldd [Hara] (ein Fork von dd, das um zahlreiche Funktionen angereichert wurde). Das Programm kann das erzeugte Image auf Wunsch splitten sowie beim Duplizieren der Datenträger automatisch Hashes berechnen (Zur Verfügung stehen MD5, SHA1/256/384/512). Es ist möglich das resultierende Image nicht nur lokal abzuspeichern, sondern auch übers Netz mittels netcat übertragen. Das Tool kann kostenfrei heruntergeladen werden und ist Bestandteil vieler Forensik-Distributionen[6] [SG]. Selbstverständlich kann auch nur das CLI-Tool *dcfldd* in Kombination mit *netcat* benutzt werden.

- **LinEn** [Abb. 5.22] - LinEn ist ein Produkt der Firma Guidance Software. Der Hauptvorteil dieser Software liegt darin, dass Datenträgerabbilder im proprietären Expert-Witness-Format abgelegt werden - resultierende Dateien sind stark komprimiert und damit deutlich kleiner als Raw-Images [Sofb].

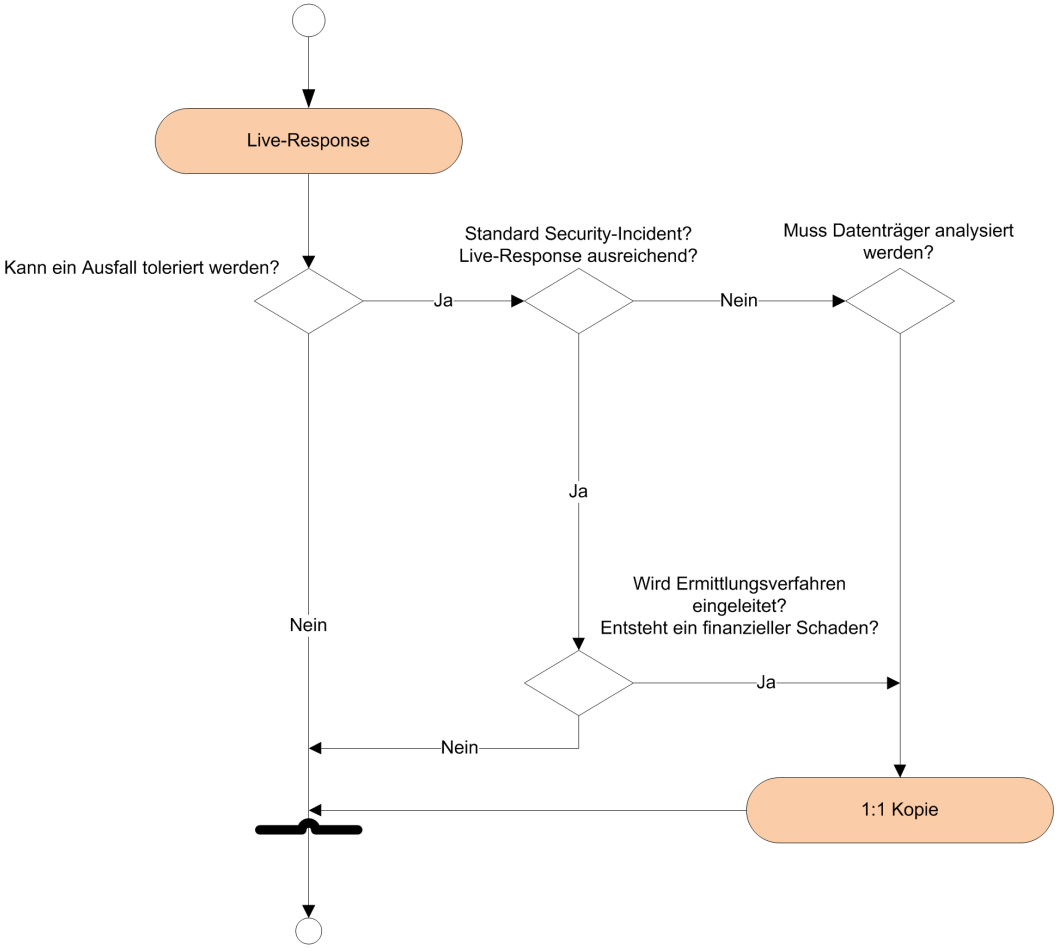

Abbildung 5.20.: Wichtige Fragestellungen bei der Gewinnung eines forensischen Duplikats.

Beide Tools arbeiten auf Bit-Ebene - der gesamte Datenträgerinhalt wird gesichert, Leserfehler werden erkannt und durch definierte Bitmuster ersetzt, reservierte Datenträgerbereiche werden ausgelesen und das erstellte Image wird durch geeignete Verfahren (z.B. MD5) verifiziert.

[6] Beispielsweise CAINE (Computer Aided INvestigative Environment), eine abgewandelte Ubuntu-CD, die speziell für Computer-Forensik-Einsatz ausgelegt worden ist. Zum Zeitpunkt der Erstellung des vorliegenden Ausarbeitung ist die aktuellste Ausgabe der Distribution 2.5.1 und kann entgeltfrei heruntergeladen und benutzt werden. Das System arbeitet von einem schreibgeschützten Medium und schließt Kompromittierung der systemeigenen Dateien von vornherein aus.[Bas]

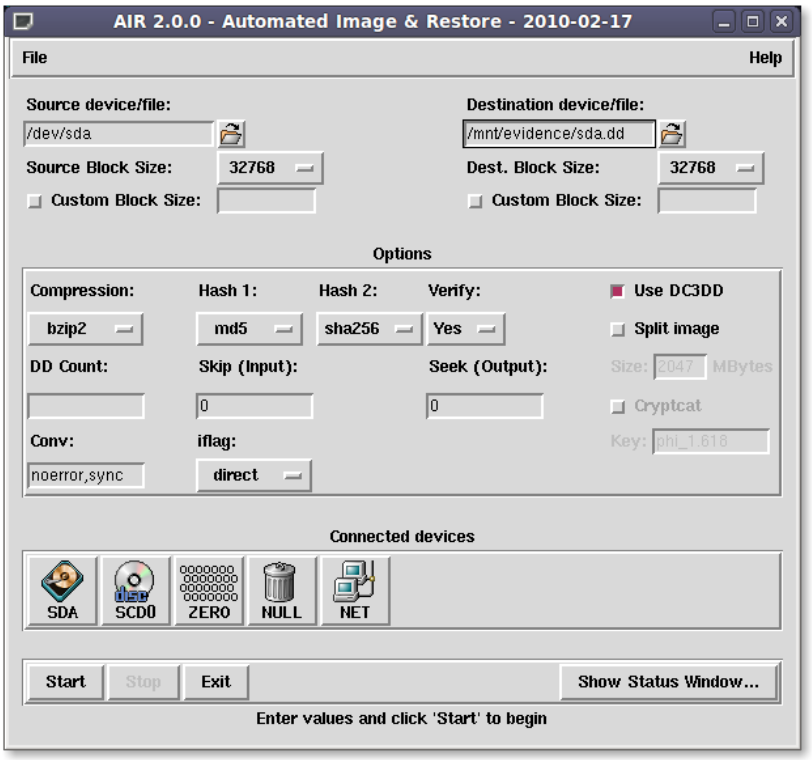

Abbildung 5.21.: AIR: Auswahl der zu duplizierenden Festplatte

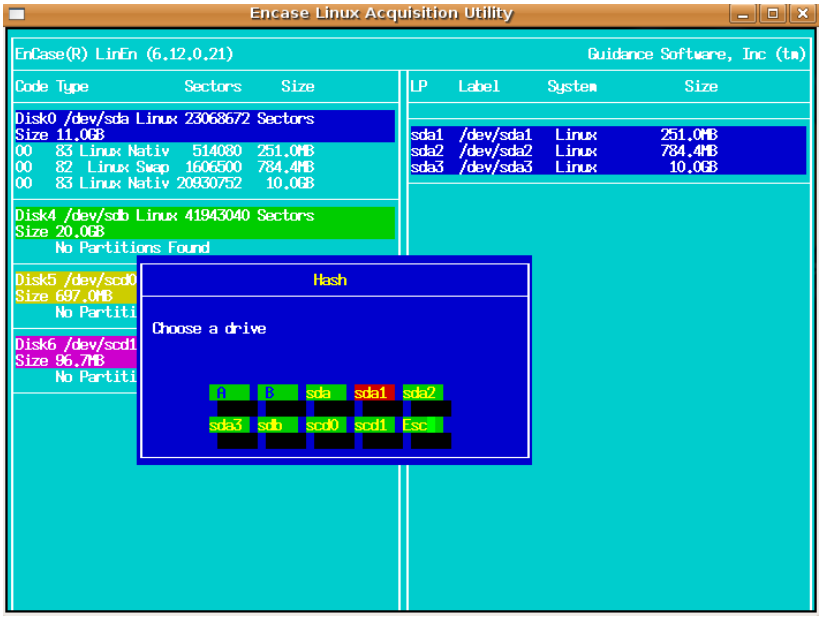

Abbildung 5.22.: Guidance Software LinEn: Auswahl der zu duplizierenden Festplatte

5.8.4. Verfahrensweisen zur Erstellung eines forensisches Duplikats

Die Verfahrensweise, wie die forensische Duplizierung am Leibniz-Rechenzentrum durchzuführen ist, wird von der allgemein anerkannten Methodik bestimmt. Allerdings müssen Besonderheiten der IT-Infrastruktur sowie bestehende Arbeitsabläufe berücksichtigt und die Verfahrensweise der Datensicherung entsprechend

angepasst werden. Außerdem gilt es sogar noch eine Unterscheidung für physische/virtuelle IT-Systems vorzunehmen, da sich hier leicht unterschiedliche Datensicherungsstrategien ergeben.

5.8.4.1. Forensische Duplikation einer virtuellen Maschine

Nach der derzeitigen Einschätzung ist es absehbar, dass die Ermittler in den meisten Fällen mit virtuellen Maschinen auf Basis von ESXi zu tun bekommen - immer mehr Systeme werden am Leibniz-Rechenzentrum virtualisiert. Die Virtualisierung führt zur effizienteren Ressourcennutzung, die messbar ist - Aufgaben können mit gleichbleibender Qualität abgearbeitet werden. Für die forensische Datenerfassung bedeutet das neue Herausforderungen - virtuelle Festplatten werden meist zentral auf einem File-Server gespeichert. Eine Image-Datensicherung eines mehrere Terabyte großen File-Servers erscheint wenig praktikabel und unwirtschaftlich.

Die Erfahrung zeigt, dass die bequemste Möglichkeit ein forensisches Duplikat einer virtuellen Maschine anzufertigen darin besteht, virtuelle Datenträger - vorliegend im VMDK-Format - zu duplizieren. Folgende Datensicherungsszenarien lassen sich aufzeigen:

- ein steriler, virtueller Datenträger wird an das kompromittierte System angeschlossen,
- Inhalt des Datenträgers wird über Netz mittels netcat an ein Network-Share übertragen,
- betroffene VMDK-Dateien werden auf das Analysesystem übertragen und analysiert.

Unabhängig davon, welche Variante gewählt wird, müssen relevante Rahmenbedingungen und Voraussetzungen erfüllt sein oder erfüllt werden. Ein Zusammenspiel mit bestehenden Strukturen und Prozessabläufen ist unbedingt notwendig - abteilungsübergreifende Zusammenarbeit ist zum Teil gefordert.

Bei der ersten Variante wird einer virtuellen Maschine eine zusätzliche Festplattendatei zugeordnet - für die forensische Duplikation macht es nämlich keinen Unterschied, ob die Zieldatei auf eine physikalische oder eine virtuelle Festplatte geschrieben wird. Auf prozesstechnischer Seite verhindern die LRZ-Sicherheitsrichtlinien eine entschiedene Vorgehensweise: Die Erstellung und das Einhängen eines neuen Laufwerks mit ausreichender Kapazität kann nur durch Administratoren erfolgen - der Administrationsaufwand kann unverhältnismäßig hoch werden und der Zeitaufwand für die Duplizierung steigt proportional zur Anzahl der involvierten Schnittstellen.

Die zweite Variante sieht die Verwendung eines zentralen, zugriffsbeschränkten Network-Shares vor. Die Größe des Speichers ist so zu wählen, dass jederzeit ausreichend Speicherplatz für Festplattenabbilder zur Verfügung steht. Der Abstimmungsbedarf mit anderen Abteilungen ist geringer, die Anzahl der Schnittstellen wird reduziert.

Unabhängig davon, welche Vorgehensweise gewählt wird, muss, bevor die Duplizierung durchgeführt werden kann, die bootfähige Live-CD-Datei vom zuständigen LRZ-Administrator eingehängt werden. Anschließend wird die Boot-Reihenfolge angepasst und die forensische Analyseumgebung gestartet. Über das einfach strukturierte Startmenü gelangt man zu den einzelnen Programmen für die Erstellung einer 1:1-Kopie. Hier bietet Air entscheidende Vorteile. Das Programm ist gut strukturiert und lässt sich einfach bedienen. Nachdem die wesentlichen Informationen eingegeben, Quellendatenträger und der Speicherort ausgewählt wurden, kann die Datenaquisitionvorgang mit einem Klick auf „Aquire“ gestartet werden, dieser sollte (abhängig von der Laufwerksgröße, der gewählten Verifikationsmethode und der Geschwindigkeit der LRZ-Infrastruktur) nach kurzer Zeit abgeschlossen sein. Die wichtigsten Optionen sind bereits voreingestellt und könnten lediglich um einen zusätzlichen Aufrufparameter „*sync*“ erweitert werden - dabei werden beim Kopiervorgang unleserliche Bereiche mit Nullen überschrieben. So lassen die Bereiche, die nicht ausgelesen werden konnten, für die spätere Betrachtung kenntlich gemacht [inu].

Der Vollständigkeit halber sei die dritte Möglichkeit erwähnt ein forensisches Duplikat einer VM anzufertigen - einige kommerzielle Produkte, allen voran Encase, können VMware Virtual Machine Disk Images einlesen und interpretieren [Sof06]. Die der kompromittierten Maschine zugewiesene virtuellen Festplatten müssen mit geeigneten Maßnahmen unter Berücksichtigung der forensischen Gesichtspunkte (Berechnung der Hashwerte) auf das Forensik-System des Ermittlers zur Analyse übertragen werden [dig].

5.8.4.2. Forensische Duplikation einer physischen Maschine

Forensische Duplikation einer physischen Maschine ist nahezu, wenn auch nicht vollkommen, identisch zur Anfertigung einer 1:1 Kopie eines virtuellen Systems.

Die technischen Möglichkeiten, forensische Datensicherungen durchzuführen, sind vielfältig - folgende Möglichkeiten sind für die Festlegung einer Verfahrensweise für die Datensicherung eines physischen Hosts denkbar:

- verdächtige Festplatte wird ausgebaut und an das Analysesystem angeschlossen,
- ein steriler Datenträger wird an das kompromittierte System angeschlossen,
- Inhalt des Datenträgers wird über Netz an ein Network-Share übertragen.

Aufgrund der komplexen IT-Infrastruktur, der Verwendung der perfomanten RAID-Verbunde und des dadurch bedingten hohen Speicherbedarfs sowie der Verfügbarkeitsanforderungen, erscheint nur Übertragung der Daten über Netz sinnvoll - immerhin gilt es diese Faktoren im forensischen Datensicherungskonzept zu berücksichtigen, welches gleichzeitig unter Kostengesichtspunkten wirtschaftlich vertretbar sein soll.

Der eigentliche Dupliziervorgang ist relativ simpel: eine Live-CD-Umgebung (z.B. CAINE Linux) kann sowohl von CD als auch von USB-Flash-Medien mit Schreibschutz gestartet werden - wobei erstere aufgrund der höheren Zugriffszeiten langsamer bootet und beim Start eines Programms zu Verzögerungen kommt. Nach dem Bootvorgang, der nach kurzer Zeit abgeschlossen sein sollte, kann ein Datenaquisitionstool (i.d.R. AIR) aufgerufen werden. Die Bedienung der Software ist weitgehend selbsterklärend und wurde bereits im Abschnitt 5.8.4.1 angeschnitten.

5.8.4.3. Einsatz eines Write-Blockers

Führende Spezialisten der Computer-Forensik legen bei forensischer Duplizierung den Einsatz eines sog. Write-Blockers [5.23] nahe - das gilt in erster Linie dann, wenn die Tragweite eines Sicherheitsvorfalls sehr weitreichend sein sollte und es absehbar ist, dass das Leibniz-Rechenzentrum eine zivil- bzw. strafrechtliche Klärung des Vorfalls anstreben wird:

> *Ein Write-Blocker (oder auch Write Protect (WP)) verhindert Schreibzugriffe auf die zu untersuchende Festplatte auf Hardware-Ebene, er wird zwischen Datenträger und Controller eingesetzt. Damit wird bei einer forensischen Untersuchung sichergestellt, dass die Integrität der Daten auf der Festplatte in keinster Weise beeinträchtigt wird.*

Die große Schwäche der Hardware-Lösungen liegt in den Bereichen Flexibilität und Kosten - beides Grundanforderungen an das vorliegende Forensik-Leitfaden. Das Hauptproblem: Durch die wachsende Datenmengen kommen am LRZ komplexe Raid-Verbunde zum Einsatz. Da die aktuell erhältliche Produkte keine Raid-Funktionalität anbieten, müssten Festplatte einzeln dupliziert werden - anschließend würde das Dateisystem in einem komplizierten Verfahren durch Einsatz von geeigneter Software (z.B. forensische Toolsammlung *pyflag* [pyf]) aus einzelnen Abbildern rekonstruiert werden. Damit steigt der Arbeitsaufwand und die Kosten.

Eine Alternative zu Hardware-Write-Blockern sind softwarebasierte Lösungen. Für Windows-Betriebssysteme sind kommerzielle Produkte von führenden Herstellern erhältlich, z.B. FastBloc Software Edition (SE) von Guidance Software [Sofc]. Unter Linux ist es mit Bordmitteln möglich, nur lesend auf eine Festplatte zuzugreifen (*mount -r*) - forensische Live-CD-Umgebungen hängen lokale Datenträger bereits standardmäßig nur im Lesemodus ein. So ist man nicht auf bestimmte, begrenzte Hardware-Lösungen angewiesen und kann mit deutlich größerer Flexibilität arbeiten. Ein wesentlicher Nachteil dieser Vorgehensweise ist jedoch, dass die Beweissicherung vor Gericht in Zweifel gezogen werden könnte - es müsste 100-prozentig sichergestellt werden, dass die Daten auf der Festplatte in keinster Weise verändert werden. Hier bietet nur der Einsatz einer Hardware-Lösung die größtmögliche Beweishärte [MS11].

Bei diesen Überlegungen darf allerdings nicht außer Acht gelassen werden, dass ein Ermittlungsverfahren nur bei einem Bruchteil aller untersuchten Fälle eingeleitet wird - nach der derzeitigen Einschätzung dürfte die Quote bei weniger als 10% liegen. Das Augenmerk der forensischen Untersuchungen liegt in erster Linie auf Quantifizierung des entstandenen Schadens sowie Feststellung der Schwachstelle(n) bzw. der Methode(n), die

zur Kompromittierung des Systems geführt haben. Basierend auf Ermittlungsergebnissen können anschließend geeignete Maßnahmen eingeleitet werden, um Sicherheitsrisiken zu eliminieren bzw. um Gefahr eines erneuten Systeimeinbruchs zu reduzieren. Angesichts dieser Überlegungen dürfte die softwarebasierte Lösung ausreichend sein.

Abbildung 5.23.: Ein Write-Blocker unterdrückt Schreibzugriffe auf Datenträger.

Quelle: [Tab]

5.9. Rechtliche Voraussetzungen und Grundlagen

Computersysteme sind längst zum Massenmedium geworden. Datenverarbeitungssysteme werden immer raffinierter, sie erlauben viele Daten zu sammeln, um sie anschließend vielfältig auszuwerten. Deshalb wird die Erfassung und Weiterverarbeitung personenbezogener Daten durch zahlreiche Datenschutzgesetze geregelt. Die Grundprinzipien des Datenschutzes sind im §3 des Bundesdatenschutzgesetzes [Bunb] präzisiert - sie müssen beim Umgang mit personenbezogenen Daten beachtet werden:

Datenvermeidung und Datensparsamkeit - Personenbezogene Daten sollen nach (technischer) Möglichkeit nicht erhoben werden. Gleichzeitig soll das Sammeln von personenbezogenen Daten auf das absolut erforderliche Minimum reduziert werden.

Anonymisierung und Pseudonymisierung - Daten der Betroffenen sollen in anonymisierter oder pseudonymisierter Form erfasst und verarbeitet werden. Ein Rückschluss auf die wahre Identität wird damit verhindert.

Rechtsicherheit steht für das Leibniz-Rechenzentrum an erster Stelle und das nicht nur weil Verletzung des Datenschutzes einem Unternehmen teuer zu stehen kommen kann - für die juristisch korrekte Sicherstellung von digitalen Spuren ist entscheidend, staatliche Auflagen des Datenschutzes zu beachten. Ihre Einhaltung ist durch technische Maßnahmen gewährleistet:

- IP-Adressen werden anonymisiert. Das bedeutet, dass das letzte Oktett einer Adresse verfremdet wird - dabei kann von einer ausreichenden Anonymisierung ausgegangen werden.
- Logdateien werden lediglich über einen Zeitraum von sieben Tagen aufbewahrt - danach erfolgt eine Löschung.

Die Implementierung der datenschutzkonformen Überwachungsprozesse bedeutet natürlich Rechtssicherheit für das Leibniz-Rechenzentrum auf der einen Seite. Auf der anderen Seite trägt der sachgerechte Umgang mit personenbezogenen Daten dazu bei, dass tatsächliche Täterermittlung nach verstreichen eines relativ knapp bemessenen Zeitfensters erschwert wird.

5.9.1. Lückenlose Dokumentation

Um Verwertbarkeit der Beweise vor Gericht zu gewährleisten, ist es für das LRZ-CSIRT sowie andere involvierte Personen äußerst wichtig und unabdingbar, einzelne Schritte genau zu dokumentiert (Chain of Custody, [5.9.2]).

Bei der Überlegungen zur Methodik der Beweissicherung wurde die Spezifik der LRZ-IT-Infrastruktur analysiert und anschließend ein stringentes methodisches Vorgehensmodell vorgeschlagen - jedes Beweisstück, jede getätigte Aktion und die verwendeten Untersuchungswerkzeuge sollen festgehalten werden. Dabei spielen eine Vielzahl von Faktoren eine wichtige Rolle, z.B. Aufbau der Infrastruktur, Dichte der virtuellen Maschinen, eingesetzte Technologien, Umgang mit rechtlichen Auflagen.

Bei der Planung und Umsetzung von Beweissicherungsrichtlinien müssen zwei Beweisarten unterschieden werden:

- Sachbeweise
- digitale Beweisspuren

In der Mehrzahl der Fälle werden Ermittlungen in erster Linie digitale Beweise hervorbringen. Eine detaillierte - auf die Gegebenheiten am LRZ abgestimmte - Vorgehensweise wird auf den folgenden Seiten vorgestellt. Da die Gewährleistung der größtmöglichen Beweishärte bei der Sicherung der Sachbeweise eine Aufgabe ist, die in der Fachliteratur bereits mehrfach angeschnitten wurde und Erfahrungsberichte vorliegen (z.B. [CP03], [Ges10]), ebenfalls sichergestellt werden soll, wird ein entsprechendes Prozedere in knapper Form vorgestellt [5.9.2.2]. Dieses Detail kann später von außerordentlicher Bedeutung sein - auch wenn die Wahrscheinlichkeit der Notwendigkeit der Erfassung der physischen Umgebung verschwinderisch gering sein dürfte.

5.9.2. Beweise & durchgeführte Aktionen dokumentieren

In den folgenden Unterabschnitten werden begriffliche Grundlagen bereitgestellt und gleichzeitig eine Reihe von Ideen und Konzepten für richtige Beweissicherung erläutert. Besonderer Wert wird auf eine strukturierte Dokumentation gelegt - erst damit ist man in der Lage, ausführliche forensische Auswertung durchzuführen und, bei Bedarf, eine Klärung vor Gericht anstreben. Um eine effektive Beweisdokumentation einzurichten, sind eine Reihe von Schritten zu durchlaufen. Nachfolgend wird das Maßnahmenbündel für den Bereich „Beweisdokumentation“ vorgestellt, das am Leibniz-Rechenzetrum systematisch eingeführt werden soll.

Bei allen grundlegenden Maßnahmen zur Spurensicherung sollen folgende Punkte beachtet werden:

- **Chain of Custody** Ein bei forensischen Untersuchungen erhaltener „Beweis“ muss manipulationssicher gespeichert und für Unberechtigten unzugänglich aufbewahrt werden. Es sollte zu jedem Zeitpunkt erkennbar sein, wer und wann im Verlauf der Untersuchung Zugriff auf die Beweise hatte.
- **Validierung der Beweise** Die Integrität der Daten ist sicherzustellen. Es ist nachweisbar sicher zu stellen, dass während der gesamten Untersuchung die Originalität der Daten erhalten bleibt. Hierfür soll von jedem elektronischen Beweis eine digitale Prüfsumme erstellt und aufbewahrt werden.
- **Das Vier-Augen-Prinzip** Für die effektive Behandlung von Beweisen ist entscheidend, dass für alle wesentlichen Feststellungen Zeugen hinzugezogen werden, die durchgeführte Aktionen verifizieren können, damit bei einer späteren juristischen Ermittlung keine Zweifel aufkommen.
- **Beschriftung der Beweise** Jeder gefundene Sachbeweis sollte beschriftet und auf einem Beweiszettel vermerkt werden [5.9.2.2].

5.9.2.1. Digitale Beweise

Das ordnungsgemäße Härten von digitalen Beweisen ist weder arbeitsintensiv noch zeitaufwändig. Für die Sicherstellung der Integrität von gesammelten Daten im Bereich der Informationstechnologie hat sich in den letzten Jahren die Verwendung von kryptografischen Verfahren (Checksummen oder Hash-Algorithmen) durchgesetzt. Wichtig ist, dass dies frühzeitig geschieht - so werden versehentliche oder beabsichtigte Manipulationen von vornherein ausgeschlossen.

Die notwendigen Prozessschritte sind schnell und effizient und lassen sich zum Teil automatisieren. Es ist absehbar, dass die Standardvorgehensweise bei einem Sicherheitsvorfall sich nicht grundlegend verändern wird: für Daten, die durch Live-Response-Toolkits gesichert werden, werden automatisch Prüfsummen berechnet, für alle weiteren sichergestellten Dateien muss dies händisch erfolgen. Das Tagesgeschäft der CSIRT-Forensiker wird um die Funktion des „Sachwalters“ erweitert: es obliegt ihnen Beweise, die auf einem File-Share im Intranet abgelegt werden, entgegenzunehmen, zu erfassen und bei Bedarf Prüfsummen zu berechnen. Die endgültige Sicherung der Beweise erfolgt auf einem zugriffsgeschützten File-Share - nur Mitglieder des CSIRT sollen Zugriff darauf haben.

Vier Beweisarten sind zu nennen, nämlich:

- Daten, die durch ein Live-Response-Toolkit erstellt wurden,
- Daten, die während der tiefergehenden Analyse vom betroffenen System sichergestellt wurden,
- forensische Duplikate bzw. bei virtuellen Maschinen virtuelle Festplatten,
- Arbeitsspeicherabbilder bzw. bei virtuellen Maschinen VMEM-Dateien.

Bei Bearbeitung von Sicherheitsvorfällen am LRZ soll ferner durch Mitglieder des CSIRT eine grundlegende, standardisierte Dokumentation aller durchgeführten Aktionen angefertigt werden. Idealerweise sollten Analyseergebnisse nicht nur im Security Incident Ticket (im iET ITSM) hinterlegt werden, sondern auch an einer zentralen Stelle (dem „Forensik-File-Share“) gesichert werden - CSIRT-Hotliner haben dafür Sorge tragen, dass die benötigten Informationen rechtzeitig eingepflegt werden. Zu einer solchen Dokumentation gehören nicht nur eine Protokollierung der durchgeführten Aktionen inklusive der Zeitpunkte unter Nennung der handelnden Personen, sondern auch der Grund, warum diese Schritte durchgeführt wurden und welche Erkenntnisse man sich davon erhofft [5.3]. Nur eine lückenlose Protokollierung aller Aktivitäten erlaubt eine Reproduktion der Untersuchungen bzw. deren Ergebnisse durch Dritte und stellt die Glaubwürdigkeit der Ermittlung sicher. Die Echtheit von solchen Ermittlungsprotokollen ist ebenfalls angemessen gegen unberechtigte Modifikationen zu schützen - hierfür kann mit Prüfsummen gearbeitet werden.

#	Zeit	Aktion	MD5-Hash	Kommentar
1	15:10	date /t & time /t	fa07b0faa701cc83e afec5ccaf93c4ea	Erfassung des aktuellen Datums & der aktuellen Uhrzeit
2	15:11	netstat.exe /ano	caac8ff2240a71e43 722ec19bbbf52b5	Anzeige aller aktiven TCP-, UDP- & IP-Verbindungen

Tabelle 5.3.: Protokolierung der durchgeführten Aktionen.

Anhand dieser Protokolle lässt sich Überblick über Maßnahmen und ihre Auswirkungen behalten sowie aufgetretene Probleme auditiert werden. Eine spätere Analyse durch Mitglieder des LRZ-CSIRT bzw. einen erweiterten Personenkreis (plus System- und Netzadministratoren) erlaubt neue Richtlinien zu entwickeln und zu implementieren, damit bei einem erneuten Auftreten schneller Gegenmaßnahmen eingeleitet werden können. Sie können sich auch als hilfreich dabei erweisen, Sicherheitslücken dauerhaft zu schließen und vorbeugende Maßnahmen ausarbeiten zu können.

5.9.2.2. Sachbeweise

Um ein effektives System zur Sicherung der gefundenen Beweise einzurichten, sind eine Reihe von Schritten zu durchlaufen und Maßnahmen umzusetzen:

1. Es sollten nähere Informationen über das betroffene System festgehalten werden.

2. Die Umgebung und das System sollten fotografiert werden.
3. Sachbeweise werden mit Etiketten beschriftet.

Um die Rechtssicherheit aufrecht zu erhalten, war es notwendig, entsprechende Formulare zu formulieren, die Elemente aus verschiedenen Quellen (u.a. [CP03], [Ges10]) zusammenführen und ein eigenes Dokumentationsformat definieren. Bei der Erstellung wurde darauf geachtet eine große Detailtreue zu gewährleisten - dadurch soll die Glaubwürdigkeit der Ermittlung sichergestellt werden. Beim Umgang mit Beweismitteln ist von dem Vier-Augen-Prinzip ausgiebig Gebrauch zu machen.

Jedes gefundene bzw. sichergestellte Objekt soll mit einem Beweiszettel versehen werden. Nach Möglichkeit werden folgende Informationen erfasst und mit einer Unterschrift bestätigt [Abb. 5.25]:

- Fallnummer und Beweis-ID - alle Beweisstücke werden sequentiell nummeriert,
- Standort des Objekts,
- Datum & Uhrzeit, an dem das Objekt gesichert wurde,
- eine genaue Beschreibung des Objekts,
- Name & Unterschrift des anwesenden Zeugen,
- Name & Unterschrift des zuständigen Ermittlers.

Neben dem Beweisstück selbst kommt häufig auch der Dokumentation der Personen, die während des Ermittlungsverfahrens Zugriff auf Beweisstück hatten, eine besondere Bedeutung zu. Es wird daher chronologisch festgehalten, wer, wann, wie und aus welchem Grund darauf zugegriffen hat.

Datum		Fall #
Uhrzeit	Standort	ID
Beschreibung		
Ermittler		Signatur

Abbildung 5.24.: Beweisetikett

Jedes gefundene bzw. sichergestellte Objekt soll anschließend beschriftet oder mit einem Beweisetikett mit folgenden Informationen versehen werden, damit es zu einem späteren Zeitpunkt eindeutig identifiziert werden kann. [Abb. 5.24]:

- Fallnummer und Beweis-ID - alle Beweisstücke werden sequentiell nummeriert,
- (urspr.) Standort des Beweisstücks,
- Datum & Uhrzeit, an dem das Objekt gesichert wurde,

- eine kurze Beschreibung des Objekts.

Alle Sachbeweise sollten an geeigneter, sicherer Stelle aufbewahrt werden, die nur für einen ausgewählten Personenkreis zugänglich ist. Dabei steht insbesondere der Schutz der Daten gegen Verfälschung und Verlust im Vordergrund. Ein Beispiel wäre ein Datentresor, zu dem nur Mitglieder des LRZ-CSIRT einen Schlüssel haben. Es würde sich anbieten, die gesamte Archivierung von Beweismitteln als LRZ-spezifisches Prozess zu beschreiben.

Zusammenfassend kann festgestellt werden, dass der Aufwand für die Beweissicherung auf den ersten Blick etwas unpraktisch erscheinen mag. Trotzdem ist die Beachtung der Richtlinien für angemessene und funktionstüchtige Beweissicherung für alle fallrelevante Objekte wichtig, damit später keine Zweifel an Herkunft und Unversehrtheit aufkommen. Bei der Dokumentation der Beweise stehen die Faktoren Zeit und Komplexität im einem direkten Verhältnis zueinander. Dieses Bewusstsein muss vorhanden sein, damit die vollständige Beweissicherung nur dann durchgeführt wird, wenn das Wesen des Sicherheitsvorfalls und mögliche Tragweite es rechtfertigt.

5.9.3. Fehler bei der Beweismittelsicherung vermeiden

Die Grundvoraussetzung in der IT-Forensik ist der korrekte Umgang mit Beweismitteln. Diese müssen entsprechend geschützt und unter forensischen Gesichtspunkten erhoben werden. Um das realisieren zu können, müssen mögliche Fehler, oftmals bedingt durch die relevanten Schnittstellen zum Incident-Response-Prozess, betrachtet werden. Ganz bewusst wird auf eine sehr ausführliche Behandlung der einzelnen Punkte verzichtet, da sie durch die anderen Unterkapiteln bereits angeschnitten wurden. Fehler bei der Beweiserhebung können, soviel sei vorweggeschickt, einfach vermieden werden, zumindest in der Theorie. In der Praxis muss im Zusammenspiel mit dem Security-Incident-Response-Team entschieden werden, ob die Tragweite der Sicherheitsvorfalls längere Downtime rechtfertigt oder aber ob sofortige Wiederherstellung der Funktionstüchtigkeit des betroffenen Computer-Systems bzw. der IT-Infrastruktur im Vordergrund steht und die Beweismittelsicherung nur zweitrangig ist.

BEWEISZETTEL

Datum		Fall #
Uhrzeit	Standort	ID

Beschreibung

Ermittler	Zeuge
Unterschrift Ermittler	Unterschrift Zeuge

Chain of Custody

Herausgabe durch	Datum/Uhrzeit	Grund	Empfangen durch
Name:			Name:
Unterschrift:			Unterschrift:
Name:			Name:
Unterschrift:			Unterschrift:
Name:			Name:
Unterschrift:			Unterschrift:
Name:			Name:
Unterschrift:			Unterschrift:
Name:			Name:
Unterschrift:			Unterschrift:
Name:			Name:
Unterschrift:			Unterschrift:
Name:			Name:
Unterschrift:			Unterschrift:

Abbildung 5.25.: Beweiszettel

6. Forensische Analyse im Fokus

Es existieren eine Vielzahl von Publikationen zum Thema forensische Datenanalyse. Umso wichtiger ist daher, dass man sich mit den gängigen Techniken kritisch und konstruktiv auseinandersetzt. Das folgende Kapitel gibt einen Überblick über verschiedene Maßnahmen der Datenanalyse für Windows Server 2008R2 und SLES 11 Systeme. Dabei besteht beispielsweise Klärungsbedarf: „Wie funktioniert Hauptspeicheranalyse?“ oder „An welchen Stellen eines Betriebssystems lassen sich Auffälligkeiten bzw. Gebrauchsspuren lokalisieren?“. Erprobte Vorgehensweise und Handlungsmuster werden schrittweise vorgestellt und sollten zu einem erfolgreichen Gelingen einer Ermittlung beitragen.

Im Verlauf dieses Kapitels werden nicht nur reine Techniken für Post-Mortem-Analyse vorgestellt, einige Maßnahmen lassen sich auch während der Live-Response anwenden. Dies betrifft besonders die Auswertung der Logdateien.

6.1. Grundlagen der Datenspeicherung

Nur eine konsequente und gut durchdachte Datenaufbewahrungs-Strategie bietet ausreichenden Schutz für im Kapitel 5 sichergestellten Spuren. Dieser Abschnitt stellt grundlegende Techniken für forensisch korrekte Datenspeicherung vor.

Die Vorgehensweise erfolgt dabei nach einem festen Schema und läuft in zwei Stufen ab. Auf einem zentralen Forensik-Server werden Daten von den einzelnen Clients oder Servern über das Netz aufbewahrt. Die Zugriffsberechtigungen werden vom LRZ-CSIRT-Team gepflegt und sollten nur die Mitglieder des Teams umfassen. Anzahl der Zugriffe auf das Beweismaterial, Anforderungen der Beweiskette (Chain of Custody) sowie Schutz der Integrität der Beweismittel sind wichtige Faktoren und gehören zu den Pflichtaufgaben des LRZ-CSIRT-Teams.

Im ersten Schritt werden Daten, die während der Datensicherungsphase bzw. während der forensischen Duplikation erzeugt wurden, auf einem zusätzlichen File-Share zwischengespeichert. Zuständiger CSIRT-Forensiker, der in diesem Fall auch als Sachwalter (vgl. Abschnitt 4.4) agiert, muss dafür Sorge tragen, dass für jede Datei Prüfsummen berechnet werden. Im zweiten Schritt werden alle Dateien auf den zentralen Datenserver verschoben.

In aller Regel genügt es nicht, nur eine Kopie der gesicherten Spuren bereitzuhalten. Da man beispielsweise immer damit rechnen muss, dass beim Analysieren ab und zu Fehler unterlaufen und Daten beschädigt werden könnten oder dass man vorsätzlich bei bestimmten Untersuchungsschritten gänzlich alle MAC-Zeiten zerstört (z.B. bei der Suche nach SUID- bzw. GID-Dateien oder Auswertung von Verzeichnis-Rechten), werden für die Analysephase Beweisspuren dupliziert. Man spricht in diesem Zusammenhang von sog. „Zweitkopien“. Dies erlaubt auch parallele Bearbeitung des Sicherheitsvorfalls durch mehrere Mitglieder des CSIRT-Teams, sofern die Priorisierung des Vorfalls dies rechtfertigt.

6.2. Wiederherstellung eines forensischen Duplikats

Der Einbindung eines forensischen Duplikats in das Analysesystem kommt eine große Bedeutung zu. Das Vorgehen beeinflusst erheblich die zum Einsatz kommenden Methoden und den Ausgang der Ermittlung. Sehr wichtig ist dabei die Berücksichtigung der Zusammensetzung der LRZ-IT-Infrastruktur - Virtualisierung spielt eine immer größer werdende Rolle. Weiterhin gilt: für forensische Untersuchungen werden oft virtuelle VMware-Datenträger gesichert - dies muss ebenfalls berücksichtigt werden.

In der Praxis hat sich das direkte Einbinden der forensischen Abbilder als logische oder physikalische Laufwerke in das Analysesystem bewährt. Diese Vorgehensweise bietet sich aufgrund der bereits bestehenden Infrastruktur und der Minimierung des Arbeitsaufwandes für die Erstellung von forensischen Abbildern für die zuständigen Administratoren [vgl. Abschnitt 5.8.4.1] an. Es existiert eine Reihe von Programmen, die diese Funktionalität bereitstellen und dafür Sorge tragen, dass der Zugriff nur lesend erfolgt: Kostenpflichtige Lösungen wie Smartmount [Datb] und Mount Image Pro [Datc] sowie das mittlerweile kostenfrei erhältliche Programm FTK Imager [Acc10], um nur einige zu nennen. Der Vorteil besteht in der Handhabbarkeit der forensischen Duplikate sowie dem direkten Einbinden von FAT- und NTFS-Partitionen als eigenständige Laufwerke - optimale Voraussetzungen für die Suche nach Malware mittels Virenscanner bzw. Anti-Rootkit-Software, da dabei das Image nicht verändert wird.

Besonders hervorzuheben ist das Programm FTK Imager [Abb. 6.1]. Es ist nicht nur kostenlos und praxiserprobt, es kann in der aktuellen Version verschiedene im professionellen IT-Umfeld gängige Dateisysteme sowie Abbildformate einbinden.

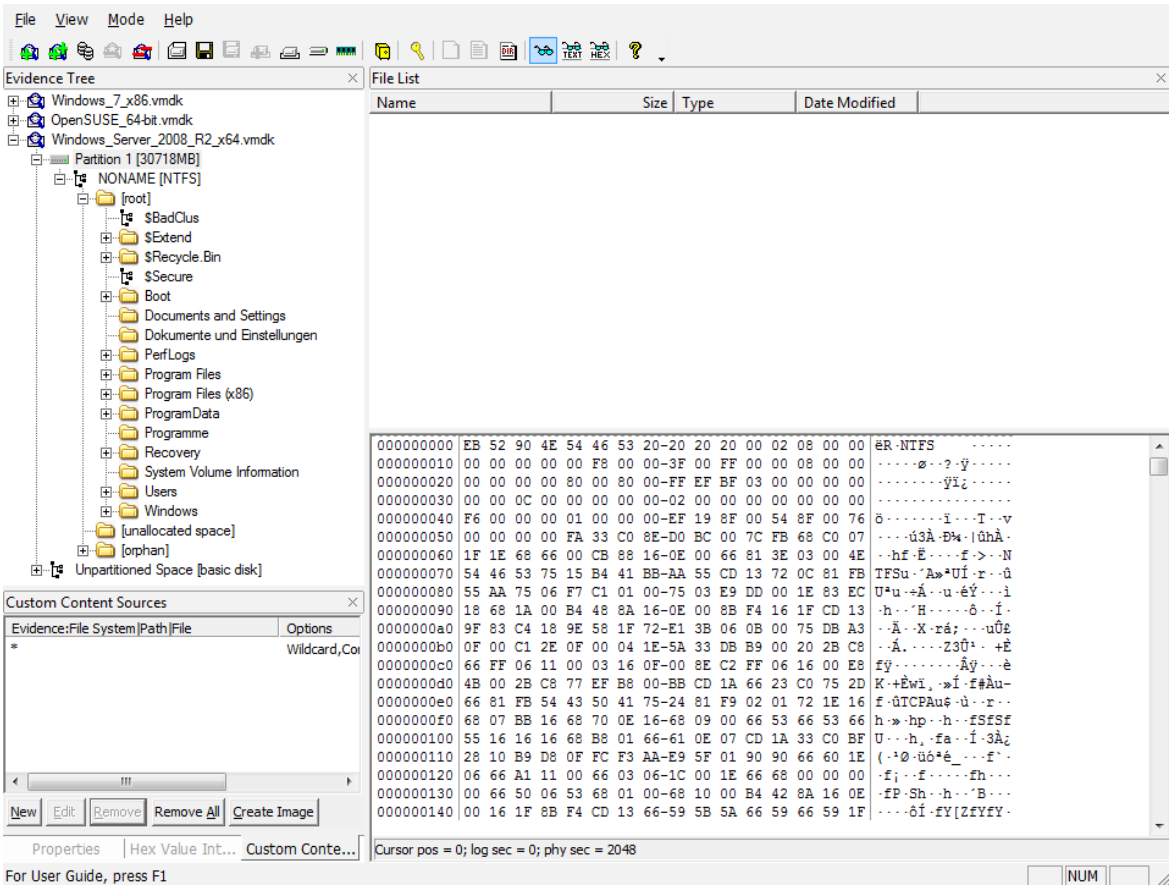

Abbildung 6.1.: Mit FTK Imager 3 lassen sich forensische Abbilder sowie virtuelle Datenträger einbinden.

Zu den unterstützten Dateiformaten gehören u.a. [Data]:

- FAT 16
- FAT 32
- NTFS
- ExFAT
- Ext2
- Ext3

- Ext4
- HFS
- HFS+
- Reiser

Wie bereits vorhin erwähnt, bietet der FTK Imager die Möglichkeit, Festplattenkopien, die mit *dd* oder anderen Werkzeugen erstellt wurden, sowie virtuelle VMware-Datenträger einzulesen. Falls das verwendete Dateisystem von Windows unterstützt wird (entweder nativ oder durch den Einsatz eines Treibers von einem Drittanbieter), können eingebundene Abbilder als normale Laufwerke verwendet werden [Abb. 6.2].

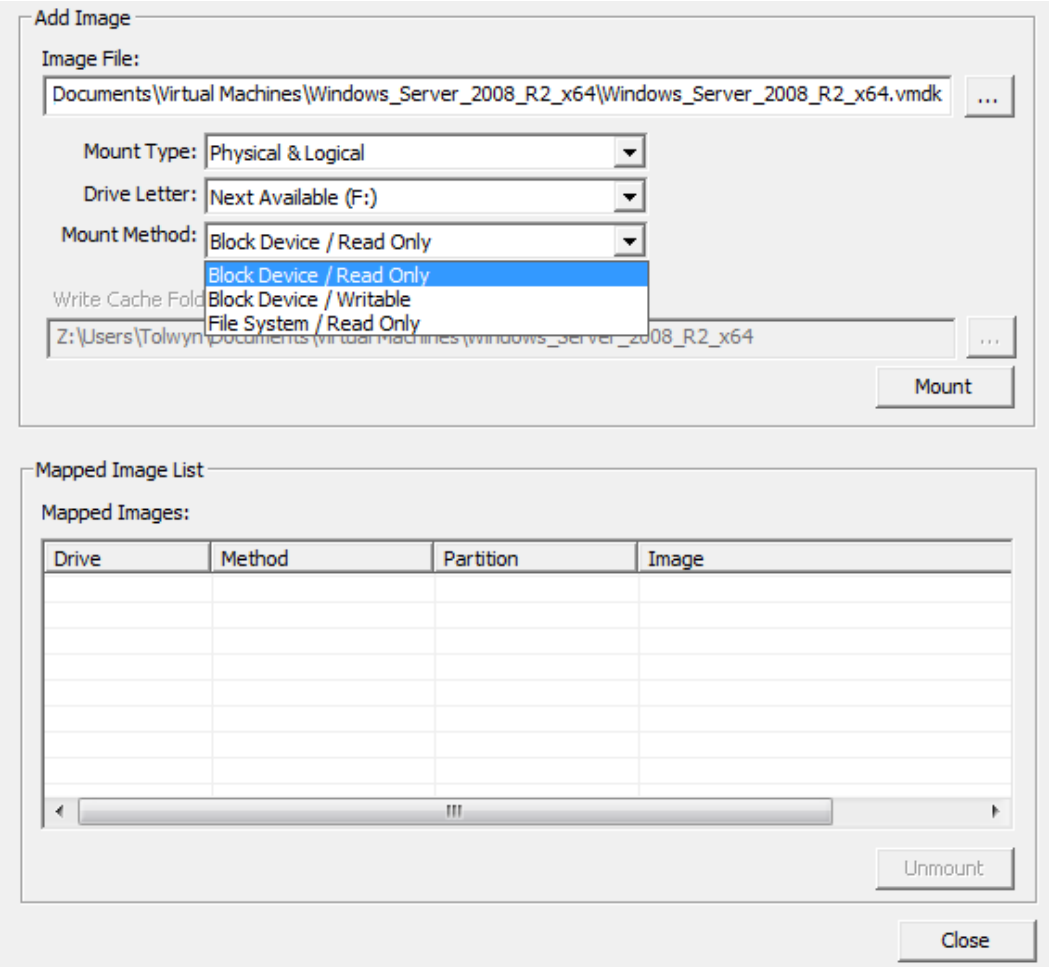

Abbildung 6.2.: FTK Imager 3: forensische Abbilder lassen sich als virtuelle physische Geräte ins System einhängen.

Besonders praktisch ist die Vorschau des kompletten Datenträgers sowie die Möglichkeit zur selektiven forensischen Extraktion von Dateien - damit erhält man wesentliche Geschwindigkeitsvorteile gegenüber anderen Lösungen, zumal man über den FTK Imager auch auf Linux-Dateisysteme unter Windows über den integrierten Dateimanager zugreifen kann, ohne diese in das Betriebssystem einhängen zu müssen [Abb. 6.3]. Hierfür waren früher zusätzliche Software-Lösungen notwendig.

Damit bringt der Einsatz des FTK-Imagers in der Praxis große Vorteile. Mit dieser Softwarelösung ist es möglich, jederzeit Festplattenkopien an das Forensik-Workstation anzubinden, unabhängig vom verwendeten Dateisystem. Dabei müssen im ungünstigsten Falle höchstens Treiber vom Dritthersteller verwendet werden.

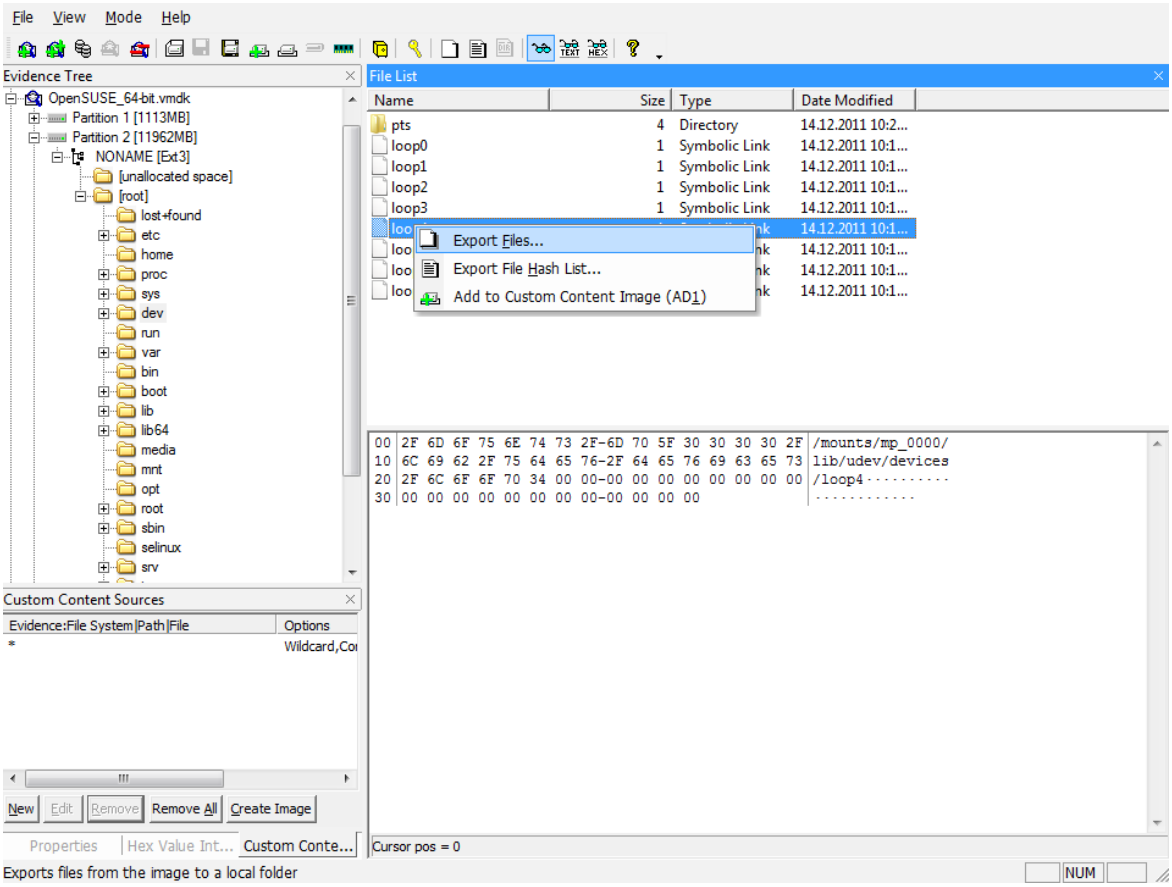

Abbildung 6.3.: FTK Imager 3: Unterstützung für diverse Linux-Dateisysteme sowie die Möglichkeit zur selektiven Extraktion von Dateien sind ebenfalls vorhanden.

6.3. Einführung in die Post-Mortem-Analyse

Wenn alle zur Verfügung stehende Daten sichergestellt wurden (siehe dazu Kapitel 5, ist für eine angemessene Analyse unter forensischen Gesichtspunkten zu sorgen. Hierzu ist es erforderlich, nach einem auf den folgenden Seiten vorgestellten Plan vorzugehen, in dem mögliche Schritte für Linux- [6.3.2] und Windows-basierte IT-Systeme [6.3.1] sowie Verantwortlichkeiten festgelegt sind. Weiterhin sollte bewusst sein, dass die hier vorgestellten Prozeduren einen Maßnahmenkatalog darstellen - abhängig vom Untersuchungsfall, Betriebssystem und Umfang der erfassten Daten wird eine Reihe von Maßnahmen ausgewählt und umgesetzt. Trotzdem sollten mögliche Maßnahmen, obwohl sie isoliert zum Einsatz kommen können, als Teil der ganzheitlichen Post-Mortem-Analyse betrachtet werden. Erst durch die Korrelation der Ergebnisse ergibt sich ein schlüssiges Gesamtbild, das auch vor Gericht Bestand hat.

Die generellen Ziele, deren Erfüllung nach dem Durchlaufen einer Ermittlung herbeigeführt werden soll, wurden bereits im Abschnitt 4.1 umrissen.

Nachfolgend wird die generelle Vorgehensweise für den Bereich „forensische Analyse“ vorgestellt.

- Der Verlauf der forensischen Bearbeitung eines Security-Incidents wird durch den SIC in Zusammenarbeit mit einem IT-Forensiker festgelegt und zuständige Administratoren über das geplante Vorgehen informiert. Einflussfaktoren, die bei der Tiefe der Analyse eine Rolle spielen, sind:
  - Priorisierung des Vorfalls,
  - Vorhandenes Personal.

- Bei der Festlegung der Vorgehensweise gilt es, alle zur Verfügung stehende Informationen zu berücksichtigen, nämlich Erkenntnisse aus der Live-Response-Phase, Daten der Netzüberwachung, Mitteilungen der externen CERTs, usw, um die bestmögliche Strategie zu finden.
- Die Analysephase ist sehr aufwendig und umfasst eine Reihe von durchzuführenden Maßnahmen, die in den folgenden Abschnitten vorgestellt werden.
- Die forensische Analyse beinhaltet die Hauptspeicheranalyse, die Logdatei-Analyse sowie weitere, zum Teil betriebssystemspezifische, Maßnahmen zur lückenlosen Aufklärung des Vorfalls. Eines der wichtigsten Ziele der Ermittlung ist es, sicherzustellen, dass wichtige Spuren erkannt und korrekt archiviert werden.
- Die initialen Ergebnisse sind dem LRZ-CSIRT mitzuteilen. Anschließend werden sie durch den SIC, zuständigen CSIRT-Forensiker sowie unter Umständen einen erweiterten Personenkreis ausgewertet. Falls notwendig werden vom SIC weitere Maßnahmen beschlossen, um fehlende Informationen in Erfahrung zu bringen, und durch den CSIRT-Hotliner an die zuständigen Stellen weitergeleitet. Eine ganzheitliche Betrachtung ist an dieser Stelle ausschlaggebend. Solche Maßnahmen können umfassen:
  - tiefgründige Analyse der Windows-Registrierungsdatenbanks,
  - Überprüfung des Dateisystems unter Einsatz verschiedener forensischer Techniken,
  - ...
- Sofern durch die Vorfallspriorisierung gerechtfertigt, sollen dabei alle Maßnahmen in Betracht gezogen werden, die zur lückenlosen Aufklärung erforderlich sind - dies gilt in erster Linie bei den kritischen IT-Systemen/Diensten.
- Falls festgestellt wird, dass noch weitere Systeme in der Umgebung betroffen sind, müssen entsprechende Incident Response Maßnahmen eingeleitet werden.
- Werden Spuren gesichert, so muss dies nach den Richtlinien geschehen, die im Abschnitt 5.9.2.1 festgelegt wurden. Von der Berechnung der Prüfsummen und dem Vier-Augen-Prinzip soll ausgiebig Gebrauch gemacht werden.
- Alle Maßnahmen und deren Ergebnisse müssen nach einem standardisierten Verfahren dokumentiert werden.
- Die Integrität der Berichte und der Archivierung der Beweise muss gewährleistet werden.

Es empfiehlt sich, Analysemaßnahmen regelmäßig zu verifizieren und auf Durchführbarkeit zu testen - es sollte eine gewisse Routine im Umgang mit der benötigten Software entwickelt werden, um Bearbeitungszeit zu minimieren.

6.3.1. Forensische Analyse unter Windows Server 2008R2

Dieser Abschnitt beschreibt die wesentlichen Methoden einer systematischen Port-Mortem-Analyse für Windows Server 2008R2. Anschließend werden, basierend auf den beschriebenen Analyse-Maßnahmen, die wichtigsten Aspekte in einer Sammlung von etwa 20 durchzuführenden ersten Schritte, zusammengeführt und vorgestellt.

Die drei Grundpfeiler für den Erfolg einer Untersuchung sind:

- Flüchtige Daten im Hauptspeicher,
- Inhalt der Windows-Registrierungsdatenbank,
- Ereignis-Logs.

Weiterhin lassen sich bei einem Windows Server 2008R2-basierten System grundsätzlich einige weitere Stellen identifizieren, die für die Ermittlung von Interesse sind:

- unallozierte Bereiche eines Dateisystems,

- File Slacks,
- Alternate Data Streams,
- MAC-Zeiten diverser Dateien.

6.3.1.1. Arbeitsspeicheranalyse

Gründe für die Initiierung: Suche im Hauptspeicher nach Malware und Spuren, Programmcode sowie Programmmodule extrahieren, Angriffsquelle bestimmen.

Verantwortlich für Umsetzung: IT-Forensiker

Geschätzter Aufwand: mittel bis hoch

Von besonderem forensischen Interesse ist die Hauptspeicheranalyse, denn oft kann erst die digitale Spurensuche direkt im Arbeitsspeicher zuverlässige Aussagen über den Zustand eines laufenden Windows-basierten Systems liefern. Ein wesentlicher Vorteil besteht darin, dass verschiedene Evasions diverser Schadsoftware umgangen werden. Der erste Durchbruch auf dem Gebiet der Hauptspeicheranalyse gelang im Jahr 2005. Während der „DFRWS 2005 Forensics Challenge“ gelang es einigen Experten auf dem Gebiet der Computer-Forensik, bereitgestellte RAM-Abbilder strukturiert zu analysieren [DFR]. Seitdem wurde eine Reihe von sowohl frei verfügbaren als auch kommerziellen Programmen entwickelt - Volatility [Sysb], HBGary Responder Pro [HBG] sowie Memoryze [Mad], nur um einige zu nennen.

Durch Sicherung und Auswertung von flüchtigen Daten können Ermittler mit diesen Programmen in Speicherabbildern unter anderem nach Spuren von Prozessen, Threads und Netzaktivität suchen, Programmcode sowie Programmmodule extrahieren, Registry untersuchen, usw.. Weiterhin lassen sich bereits terminierte Prozesse und Netzverbindungen nachweisen, sofern betreffende Speicherbereiche noch nicht mit neuen Inhalten belegt wurden. Dadurch wird unter gewissen Umständen eine Rekonstruktion des Tathergangs überhaupt erst ermöglicht.

Zu den Voraussetzungen für den Erfolg der Hauptspeicheranalyse gehört die Verwendung von geeigneten Werkzeugen. Die Programmauswahl für die Nutzung am LRZ ist zum Zeitpunkt der Erstellung dieses Leitfadens sehr gering. Es existiert nur ein frei verfügbares Werkzeug, nämlich das in Python geschriebenes Forensik-Tool „Volatility Framework“, welches sowohl Windows- als auch Linux-Hauptspeicherabbilder analysieren kann. Sowohl Unterstützung für 64-Bit Windows-Systeme als auch die Linux-Unterstützung sind allerdings noch im Alphastadium und daher zum Teil unzuverlässig. Laut den Aussagen der Entwickler kann mit einer stabilen Version Anfang 2012 gerechnet werden [Aut]. Unabhängig davon steht eine ganze Reihe von Plugins für verschiedene Analysen zur Verfügung, beispielsweise für die Erkennung von Malware, Prozess-Enumeration und Datenwiederherstellung, die die Arbeit deutlich erleichtern sollten [Wikb].

Die manuelle Installation von Volatility ist etwas zeitaufwändig - es gibt derzeit leider keine „out-of-the-box“-Lösung - wird aber sehr ausführlich in der offiziellen Wiki behandelt [Halb]. Es wird detailliert erläutert, welche Voraussetzungen erfüllt sein müssen, samt der expliziten Anweisungen zur Installation. Außerdem wird die Methode zur Installation sowohl für Linux- als auch Windows-Systeme erklärt.

Folgende Voraussetzungen müssen erfüllt sein:

- lokale Python-Installation, vorzugsweise in der Version 2.6 [Pyt],
- Pycrypto - Python Cryptography Toolkit [Lit],
- Distorm3 Disassembler [dis],
- Yara-Python - ein Toolkit zur Identifikation und Klassifikation von Malware [Yar],
- ein Subversion-Client, z.B. TortoiseSVN [Tor].

Subversion-Checkout erfolgt mit der folgenden URL: *http://volatility.googlecode.com/svn/* - damit erhält man Zugriff sowohl auf die Linux-Version als auch auf die Unterstützung für 64-Bit Windows-Betriebssysteme.

Weiterhin muss für eine geeignete Protokollierung während des Analysevorgangs gesorgt werden (siehe auch 5.9.2 „Beweise & durchgeführte Aktionen dokumentieren“).

Die Vorgehensweise bei der Analyse von Hauptspeicherabbildern lässt sich in mehrere Bereiche unterteilen - sie entsprechen in etwa der Aufteilung bei der Live-Response [5.5.2]. Grundlagen der Erstellung eines Speicherabbilds wurden bereits im Abschnitt 5.5.2.2 vermittelt.

Nachfolgend werden relevante Analyse-Aspekte für Windows-basierte Systeme dargestellt und an einigen Beispielen mittels Volatility Framework erklärt[1]).

Die Analyse dient der Prüfung, ob ein Einbruch erfolgreich war und welche Spuren durch einen Täter hinterlassen wurden. Die Erfahrung zeigt, dass es nicht reicht, eine Liste von Schritten sequentiell abzuarbeiten - stattdessen wird nach jedem Arbeitsschritt über die weitere Vorgehensweise entschieden. Dieses Bewusstsein muss bei den Mitglieder des LRZ-CSIRT vorhanden sein. Weiterhin ist bei den Überlegungen entsprechend zu berücksichtigen, dass mit steigendem zeitlichen Abstand zum SI die Zahl der Spuren sinken könnte. Unabhängig davon können folgende Teilbereiche der RAM-Forensik unterschieden werden:

- Untersuchung der laufenden Prozesse,
- Analyse der Netzverbindungen,
- Analyse der registrierten Dienste,
- Untersuchung der Teilbereiche der Registry,
- fortgeschrittene Malwareanalyse.

6.3.1.1.1. Prozesse

Wie schon in der Prozessbeschreibung Live-Response im Kapitel 5 erklärt wurde, ist die Auswertung der laufenden Prozesse maßgeblich, um bei Verdacht auf enthaltenen Schadcode, diesen erkennen zu können.

Windows Server 2008R2 verwaltet Informationen über Prozesse in speziellen „executive process“-Blöcken (EPROCESS), die wiederum in einer doppelt verketteten Liste geführt werden. Jedes EPROCESS-Block enthält eine LIST_ENTRY Struktur mit den Pointer „forward link“ (FLINK) und „backward link“ (BLINK) beinhaltet. FLINK und BLINK sind Zeiger, die auf den nachfolgenden bzw. vorhergehenden Prozess in der Liste verweisen. Tools wie *Tlist* [5.5.2.5] verwenden diese Liste, um aktuell aktive Prozesse zu enumerieren und aufzulisten [Gil].

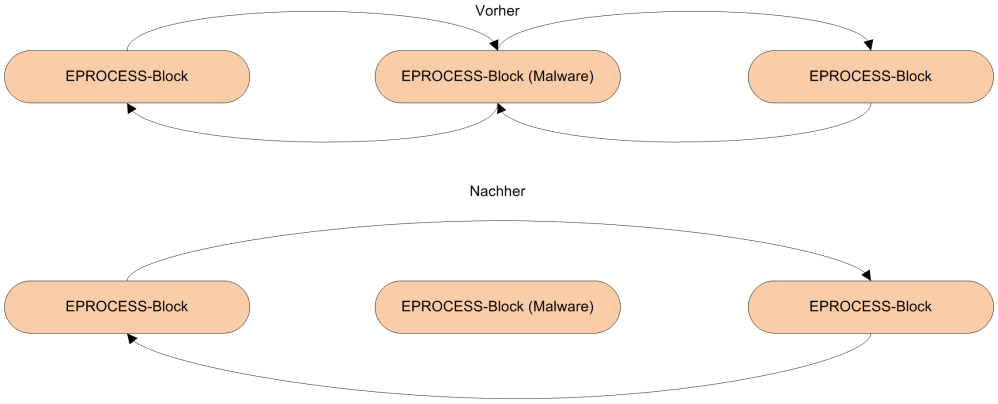

Abbildung 6.4.: Manipulation der Forward/Backward-Links in EPROCESS Blöcken

Eines des Standardvorgehensweisen bei Malware-Autoren zur Tarnung eines Prozesses ist das Auffinden und Modifizieren der EPROCESS-Struktur. Sie manipulieren direkt FLINK- und BLINK-Zeiger vorhergehender und nachfolgender Blöcke, denn bei einem Aufruf des Windows Task-Managers wird die verlinkte Liste sequentiell ausgewertet, der „versteckte“ Prozess wird nicht in der Auflistung auftauchen. In diesem Zusammen-

[1] Anders als bei der Live-Response kann die Reihenfolge variiert werden.

hang spricht man auch von „Direct Kernel Object Manipulation“ (DKOM) [GH05]. Das folgende Bild 6.4 veranschaulicht die Funktionsweise von DKOM.

Auf diese Art versteckte Prozesse können nur durch Analyse der sog. „pool tags“ [2] ausfindig gemacht werden.

Oft tarnen sich Schadprozesse als bekannte Anwendungen und/oder Dienste. Ein aktuelles Beispiel kann der Abbildung 6.5 entnommen werden. In Wirklichkeit handelt es sich bei einem der beiden dwm-Prozesse (Desktop Window Manager) um eine Version des Darkness DDoS Bots.

Name	Pid	PPid	Thds	Hnds	Time
0x85246D40:csrss.exe	388	356	9	451	2011-12-15 20:33:56
0x85BD0D40:wininit.exe	420	356	3	82	2011-12-15 20:33:58
. 0x85C30968:services.exe	536	420	7	208	2011-12-15 20:34:00
.. 0x85F18468:taskhost.exe	1536	536	8	0	2011-12-15 20:34:19
.. 0x85CFCD40:svchost.exe	648	536	9	364	2011-12-15 20:34:08
... 0x861A6748:FlashUtil11e_A	2408	648	3	76	2011-12-15 20:41:17
.. 0x862224D0:TPAutoConnSvc.	300	536	8	138	2011-12-15 20:34:35
... 0x8623D070:TPAutoConnect.	2012	300	5	153	2011-12-15 20:34:40
.. 0x840171E8:svchost.exe	1040	536	11	329	2011-12-15 20:34:13
.. 0x85CDD228:dwm.exe	2648	536	8	217	2011-12-16 11:31:28
.. 0x8409E698:svchost.exe	2992	536	13	357	2011-12-15 20:36:33
.. 0x85E9A238:svchost.exe	1316	536	17	312	2011-12-15 20:34:16
.. 0x85F1B3B8:svchost.exe	2864	536	7	120	2011-12-15 20:36:28
.. 0x85E80030:spoolsv.exe	1288	536	16	367	2011-12-15 20:34:15
.. 0x85DAE4E8:svchost.exe	820	536	18	458	2011-12-15 20:34:10
.. 0x84E2ED40:SearchIndexer.	2120	536	15	737	2011-12-15 20:34:44
.. 0x85DCC030:svchost.exe	888	536	28	1134	2011-12-15 20:34:10
.. 0x84E27B18:svchost.exe	728	536	5	270	2011-12-15 20:34:09
.. 0x85DC7030:svchost.exe	860	536	19	458	2011-12-15 20:34:10
... 0x85F34998:dwm.exe	1656	860	5	134	2011-12-15 20:34:20
.. 0x840B5D40:sppsvc.exe	2960	536	4	155	2011-12-15 20:36:32
.. 0x85F4D860:vmtoolsd.exe	1640	536	7	220	2011-12-15 20:34:20
.. 0x85E118E8:svchost.exe	1140	536	18	571	2011-12-15 20:34:13
.. 0x85EC1810:VMUpgradeHelpe	1788	536	4	93	2011-12-15 20:34:25

Abbildung 6.5.: pstree: Windows-Prozess dwm.exe taucht doppelt in der Prozessliste auf.

Die Analyse der verdächtigen Prozesse ist zudem eng verbunden mit der Auswertung der geöffneten Handles und DLL-Dateien: Verweise auf offene Dateien, Registrierungsschlüssel, Synchronisierungsprimitive, verwendete Bibliotheken und dergleichen können sehr hilfreich sein, um beispielsweise die Suche in der Registry auf einige wenige Schlüssel einzuschränken.

Volatility stellt für die Erkennung und Analyse von Prozessen eine Vielfalt an Werkzeugen bereit. Syntax jedes einzelnen Befehls lässt sich mit dem Parameter *-h* aufrufen.

Offset(P)	Name	PID	PPID	PDB	Time created	Time exited
0x04189968	System	4	0	0x00185000	2011-12-15 20:33:53	
0x2ee27b18	svchost.exe	728	536	0x2eb0e140	2011-12-15 20:34:09	
0x2ee2ed40	SearchIndexer.	2120	536	0x2eb0e440	2011-12-15 20:34:44	
0x2fc171e8	svchost.exe	1040	536	0x2eb0e200	2011-12-15 20:34:13	
0x2fc9d278	darkness_8.exe	3560	1668	0x2eb0e3a0	2011-12-16 13:30:48	2011-12-16 13:30:49
0x2fc9e698	svchost.exe	2992	536	0x2eb0e4e0	2011-12-15 20:36:33	
0x2fcb5d40	sppsvc.exe	2960	536	0x2eb0e4c0	2011-12-15 20:36:32	

Abbildung 6.6.: psscan: spürt versteckte und beendete Prozesse auf.

- **pslist** - zeigt detaillierte Informationen über Prozesse, Anzahl der Threads, Anzahl der Handles, Prozessstartzeit sowie Memory-Offset an. Kann keine versteckten oder bereits beendeten Prozesse erkennen.
- **pstree** - entspricht in der Funktionalität *pslist*, stellt laufende Prozesse allerdings in einer Baumansicht dar [6.5].
- **psscan** - findet durch die Auswertung von „pool tags“ bereits beendete oder von einem Rootkit versteckte Prozesse. Ein Beispiel kann der Abbildung 6.6 entnommen werden. Die Auflistung kann zudem mit der Ausgabe von *pslist*/*pstree* verglichen werden, um Anomalien aufzuspüren.
- **dlllist** - zeigt alle DLL-Dateien an, die zum Zeitpunkt der Erstellung des Arbeitsspeicherabbilds auf dem IT-System von gestarteten Prozessen referenziert wurden. Bei versteckten Prozessen muss mit seg-

[2] Alloziierte Speicherblöcke werden mit einem 4-byte großen Zeichenblock im ASCII-Format markiert, mit welchem sie eindeutig identifiziert werden können.

mentierten Adressen (engl. „physical offset“) der EPROCESS-Objekte gearbeitet werde, beispielsweise *–offset=0x8b5378*.

- **handles** - listet alle von einem verdächtigen Prozess verwendete Bilbiotheken, Registrierungsdatenbankeinträge und Dateien. Mit dem Befehlszeilenparameter *–silent* lässt sich die Ausgabe auf wichtige Informationen reduzieren sowie mit *–pid=Prozess-ID* auf ein Prozess beschränken.

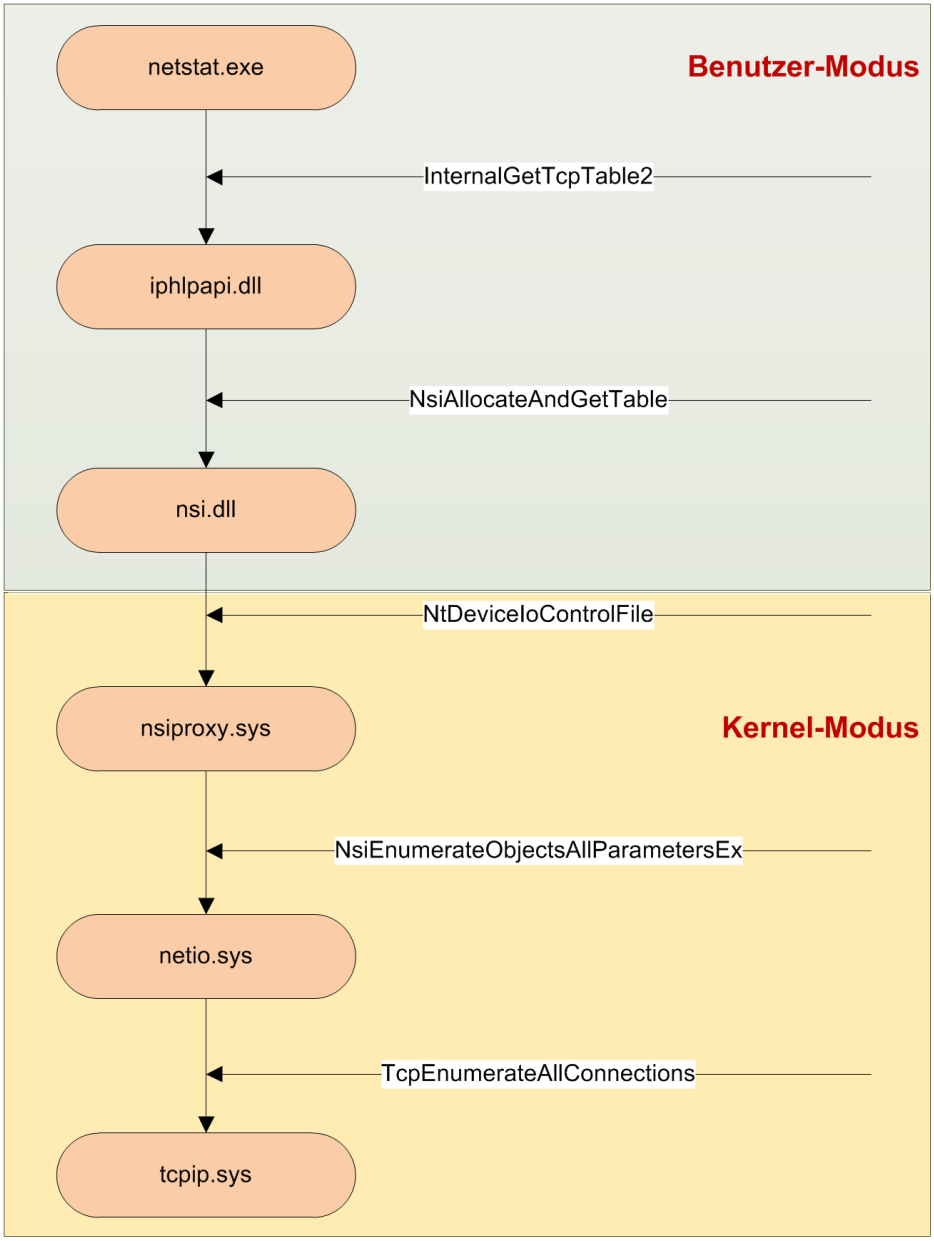

Abbildung 6.7.: netstat ruft eine Reihe von Systemfunktionen auf, die manipuliert sein könnten.

6.3.1.1.2. Netzverbindungen

Die Analyse des Arbeitsspeichers bietet die Möglichkeit, weitere Ansätze über den Verlauf des Angriffs sowie Prozessaktivitäten im Netz zu sammeln. Diese Vorgehensweise ist vor allem dann sinnvoll, wenn angenommen werden kann, dass das IT-System von einem eventuell schadhaften Code befallen wurde. Einer der wesentlichen Vorzüge einer RAM-Analyse liegt darin, dass sich nur durch diesen Analyseansatz herausfinden lässt,

welche Verbindungen zum Zeitpunkt der Abbilderstellung überhaupt aktiv waren. Oft hat man zur Laufzeit eines Systems keine Chance, Auffälligkeiten zu erkennen, da ein Kernel-Rootkit die Windows-API manipulieren und seine Verbindungen gleichzeitig verstecken könnte. Der Befehl *netstat*, der dafür in der Regel beim Live-Response verwendet wird, ist ein mächtiges Werkzeug, das alle aktiven Verbindungen sowie Verbindungen, die gerade aufgebaut oder im Abbauprozess sind, anzeigt. *netstat* bekommt die Informationen aus dem Kernel. Dabei wird eine Reihe von Systemfunktionen aufgerufen: *iphlpai.dll* und *nsi.dll* im Benutzer-Modus sowie *nsiproxy.sys*, *netio.sys* und *tcpip.sys* im Kernel-Modus - vergleiche auch Abbildung 6.7. Wenn eine der Dateien trojanisiert ist, können Informationen unvollständig oder verfälscht sein.

Genau an dieser Stelle setzt forensische Speicheranalyse an - die Windows-API wird umgangen. Ein weiterer Vorteil ist, dass bereits beendete Netzverbindungen identifiziert werden könne. Wie bei der Prozessspeicheranalyse gilt - die entsprechenden Hauptspeicherbereiche dürfen noch nicht überschrieben worden sein.

Im Folgenden geht es lediglich um die praktische Anwendung, für detailliertere Darstellungen der Funktionsweise sei auf die sehr umfangreiche Erklärung zu diesem Thema von Michael Ligh verwiesen [Ligb].

Man kann im Wesentlichen folgende Analysetypen unterscheiden:

- **generelle Suche nach Auffälligkeiten:** Finden sich beispielsweise sehr viele Verbindungsanfragen, so ist davon auszugehen, dass dieses System für einen Portscan oder einen DDoS-Angriff missbraucht wurde. Weitere Anhaltspunkte liefert die Auswertung des Portmappings.
- **Prozessanalyse:** Netzverhalten von verdächtigen Prozessen kann untersucht werden. Oft lassen sich beispielsweise Verbindungen zu C&C-Servern nachweisen.

Volatility Framework stellt hierfür den Befehl *netscan* zur Verfügung. *netscan* unterstützt sowohl UDP- als auch TCP-Verbindungendpunkte und kann sowohl mit TCPv4 als auch mit TCPv6-Verbindungen umgehen.

Ein Beispiel: Die folgende Abbildung 6.8 zeigt einen Auszug eines netscan-Protokolls. Bereits geschlossene Verbindungen zu einem bekannten *Command & Control Server* (173.194.69.191) auf Port 80 sowie ein geöffneter Socket auf Port 5005 können identifiziert und einem bereits als auffällig erkannten Prozess zugeordnet werden.

Offset(P)	Proto	Local Address	Foreign Address	State	Pid	Owner
0x2ddb5620	TCPv4	0.0.0.0:49156	0.0.0.0:0	LISTENING	536	services.exe
0x2ddb5620	TCPv6	:::49156	:::0	LISTENING	536	services.exe
0x2ddb5910	TCPv4	0.0.0.0:445	0.0.0.0:0	LISTENING	4	System
0x2ea9a9f8	TCPv4	0.0.0.0:5005	0.0.0.0:0	LISTENING	1464	580846.exe
0x2db229e8	TCPv4	0.0.0.0:49506	173.194.69.191:80	CLOSED	2852	iexplore.exe
0x2ee43df8	TCPv4	0.0.0.0:49532	192.228.79.201:80	CLOSED	2852	iexplore.exe
0x2fa13ac8	TCPv4	0.0.0.0:49540	192.228.79.201:80	CLOSED	2852	iexplore.exe
0x2fa21b08	TCPv4	0.0.0.0:49542	91.193.192.95:80	CLOSED	1464	580846.exe
0x2fc7e008	TCPv4	0.0.0.0:49530	192.228.79.201:80	CLOSED	2852	iexplore.exe
0x2fcf4458	TCPv4	0.0.0.0:49541	91.193.192.95:80	CLOSED	1464	580846.exe
0x2fdd2368	TCPv4	0.0.0.0:49519	255.255.255.255:80	CLOSED	2852	iexplore.exe
0x2da191b8	UDPv4	0.0.0.0:5355	*:*		1140	svchost.exe
0x2da9c200	UDPv4	192.168.36.134:137	*:*		4	System
0x2daf6ae8	UDPv4	127.0.0.1:1900	*:*		2864	svchost.exe

Abbildung 6.8.: netscan spürt geöffnete, versteckte und beendete Verbindungen auf.

6.3.1.1.3. Dienste

Auszuführende Dienste bieten eine weitere Angriffsfläche für Hacker. Oft werden neue Dienste mit vermeintlich regulären Namen registriert. Tauchen unerwartete neue Dienste auf, lassen sich diese aus dem Speicher auslesen. Die Analyse kann auf unterschiedlichen Ebenen ansetzen. Wichtig hierbei ist in erster Linie der vollständiger Pfad für den registrierten Dienst: bei Benutzer-Modus Rootkits (engl. „user mode“) wäre dies eine ausführbare Datei, bei Kernel-Modus Rootkits (engl. „kernel mode“) meistens ein Treiber. Einen weiteren Grundpfeiler zum Erfolg der Untersuchung tragen PID (sofern zutreffend), Dienstname, angezeigte Dienstname (engl. DisplayName) sowie der aktuelle Status bei. Solange Zugriff auf die Speicherabbilder möglich ist, sind alle diese Informationen auslesbar.

Für die Untersuchung von registrierten Diensten stellt Volatility den Befehl *svcscan* zur Verfügung, Bestandteil des malware.py-Plugins, das ursprünglich für das Malware Analyst's Cookbook [Halc] entwickelt wurde. Ein Beispiel kann dem nachfolgenden Listing 6.1 entnommen werden. Die Ausgabe wurde der Übersichtlichkeit halber um einige Informationen gekürzt.

```
Pid      Name        DisplayName           Type                         Path
2648     'darkness'  'IpSectPro service'   SERVICE_WIN32_OWN_PROCESS    C:\Windows\system\dwm.exe
-------- 'VgaSave'   'VgaSave'             SERVICE_KERNEL_DRIVER        \Driver\VgaSave
-------- 'NetBIOS'   'NetBIOS Interface'   SERVICE_FILE_SYSTEM_DRIVER   \FileSystem\NetBIOS
-------- lanmandrv   lanmandrv             SERVICE_KERNEL_DRIVER        \Driver\lanmandrv
```

Listing 6.1: svcscan liefert umfangreiche Informationen zu gestarteten Diensten

Die beispielhaften Ausgabedaten zeigen zum einen die Datei *dwm.exe* - dies impliziert zwar Desktop Window Manager, die Datei befindet sich allerdings im „falschen“ Verzeichnis - *C:\Windows\System* statt *C:\Windows\System32*. Weiterhin wird der DWM-Dienst eigentlich als *UxSms* geführt [Wikd]. Darüber hinaus kann der Auflistung der Hinweis auf den Kernel-Treiber des *Win32/Laqma*-Wurms entnommen werden [Micd].

6.3.1.1.4. Registry

Die Registry besteht sowohl aus flüchtigen als auch nichtflüchtigen Daten - zur Laufzeit des Betriebssystems wird die komplette Registrierungsdatenbank in den Arbeitsspeicher geladen. Registry-Analyse spielt bei forensischen Untersuchungen von jeher eine bedeutende Rolle - viele Potentiale müssen erst in der Praxis entdeckt werden. Es muss daher unterschieden werden zwischen den theoretisch vorhandenen Möglichkeiten und der praktischen Umsetzung. Im Verlauf dieses Kapitels werden anhand diverser Beispiele Analyseansätze vorgestellt, die allesamt den Ermittlungsprozess beschleunigen können. Einzige Voraussetzung - es liegt ein Hauptspeicherabbild vor. Die Registry-Analyse im Hauptspeicher führt nicht nur zu effizienteren und schnelleren Prozessschritten, die messbar sind - es lassen sich auch Daten analysieren, die ansonsten nicht zur Verfügung stünden.

Ganz bewusst wird an dieser Stelle auf Ausführungen zur Struktur der Registry verzichtet - entsprechendes Hintergrundwissen wird im Abschnitt 6.3.1.2.1 vermittelt.

Die Registry-Analyse verfolgt das Ziel, diejenigen Merkmale einer Malware zu identifizieren, die oft vorkommen und daher keinen unverhältnismäßig hohen Arbeitsaufwand erfordern. Dies gilt für eine Reihe von Registry-Schlüsseln, die ausführlich im Abschnitt 6.3.1.2.2 vorgestellt werden.

Die Umsetzung von Maßnahmen zur Registry-Analyse hängt von der verwendeten Software ab. Es bietet sich allerdings an, auch an dieser Stelle auf das Werkzeug Volatility zu setzen. Die grundlegenden Ansätze werden im Folgenden beschrieben und die entsprechenden Funktionalitäten der Software erläutert.

In der Praxis hat sich folgender Ansatz als praktisch erwiesen:

- Mit dem Befehl **hivelist** werden die Speicher-Offsets der zugehörigen Datenstrukturen im Hauptspeicher bestimmt. Sämtliche Analysen werden dann mit diesen Abständen durchgeführt - so lassen sich Informationen zu den einzelnen Hives extrahieren (vgl. Listing 6.2).

```
J:\Vol>python vol.py --profile=Win7SP1x64 -f ws_2008R2.vmem hivelist
Volatile Systems Volatility Framework 2.1_alpha
Virtual     Physical    Name
0x8b9a99d0  0x2cd769d0  \Device\HarddiskVolume1\Boot\BCD
0x8be9b840  0x1c935840  \??\C:\Users\Tolwyn\ntuser.dat
0x8beae580  0x1e5ae580  \??\C:\Users\Tolwyn\AppData\Local\Microsoft\Windows\UsrClass.
    dat
0x8f457008  0x2b33d008  \SystemRoot\System32\Config\DEFAULT
0x8f5d73b0  0x22f083b0  \SystemRoot\System32\Config\SECURITY
0x90e8c008  0x1a9c8008  \??\C:\System Volume Information\Syscache.hve
0x82b76140  0x02b76140  [no name]
0x82216008  0x23f03008  \SystemRoot\System32\Config\SAM
0x8235f008  0x23417008  \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0x823b8378  0x22f91378  \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0x8780c8b0  0x066898b0  [no name]
0x8781a258  0x066a3258  \REGISTRY\MACHINE\SYSTEM
0x87844008  0x006d7008  \REGISTRY\MACHINE\HARDWARE
0x8b8889d0  0x20b089d0  \SystemRoot\System32\Config\SOFTWARE
```

Listing 6.2: hivelist zeigt Memory-Offsets von Registry-Hives an

- Via **printkey** lassen sich einzelne Registry-Schlüssel anzeigen. Volatility setzt hierbei auf virtuelle Offsets, die im Vorfeld durch den Einsatz von **hivelist** bestimmt wurden (vgl. Listing 6.3). Eine Schlüsselwortsuche lässt sich entweder auf die komplette Registry oder ein bestimmtes Hive anwenden.

```
J:\Vol>python vol.py --profile=Win7SP1x64 -f ws_2008R2.vmem printkey 0x8b8889d0
Volatile Systems Volatility Framework 2.1_alpha

Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \REGISTRY\MACHINE\SYSTEM
Key name: CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144} (S)
Last updated: 2011-12-15 20:33:41

Subkeys:
  (S) ControlSet001
  (S) ControlSet002
  (S) MountedDevices
  (S) RNG
  (S) Select
  (S) Setup
  (S) WPA
  (V) CurrentControlSet
(...)
```

Listing 6.3: Via printkey lassen sich einzelne Hives anzeigen

Weiterhin kann die übliche Autostart-List (vgl. Abschnitt 6.3.1.2.2 sowie Listing 6.4) abgearbeitet werden und der Inhalt der einzelnen Einträge anzeigen lassen (siehe Listing 6.5). Ferner wird bei jedem Schlüssel *Last Write Time* [6.3.1.2.2] angezeigt - damit lassen sich Änderungen in der Registry mit anderen Indizien korrelieren.

```
J:\Vol>python vol.py --profile=Win7SP1x64 -f ws_2008R2.vmem printkey -o 0x8b8889d0 -K
    "Microsoft\Windows\CurrentVersion\Run"
Volatile Systems Volatility Framework 2.1_alpha

Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: User Specified
Key name: Run (S)
Last updated: 2011-12-15 20:32:20

Subkeys:

Values:
REG_SZ   VMware Tools    : (S) "C:\Program Files\VMware\VMware Tools\VMwareTray.exe"
REG_SZ   VMware User Process : (S) "C:\Program Files\VMware\VMware Tools\VMwareUser.
    exe"

J:\Vol>python vol.py --profile=Win7SP1x64 -f ws_2008R2.vmem printkey -o 0x8be9b840 -K
    "Software\Microsoft\Windows\CurrentVersion\Run"
Volatile Systems Volatility Framework 2.1_alpha

Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: User Specified
Key name: Run (S)
Last updated: 2011-12-26 12:44:20

Subkeys:

Values:
REG_SZ        SothinkSwx      : (S) C:\Users\Tolwyn\AppData\Local\Temp\SothinkSwx.exe
```

Listing 6.4: Via printkey lassen sich einzelne Schlüssel anzeigen

```
J:\Vol>python vol.py --profile=Win7SP1x64 -f ws_2008R2.vmem printkey -o 0x8781a258 -K
    "ControlSet001\services\dwm"
Volatile Systems Volatility Framework 2.1_alpha

Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: User Specified
Key name: dwm (S)
Last updated: 2011-12-16 13:26:29

Subkeys:

Values:
REG_DWORD     Type            : (S) 272
REG_DWORD     Start           : (S) 2
```

```
REG_DWORD     ErrorControl    : (S) 0
REG_EXPAND_SZ ImagePath       : (S) C:\Windows\system\dwm.exe
REG_SZ        DisplayName     : (S) IpSectPro service new
REG_SZ        ObjectName      : (S) LocalSystem
```

Listing 6.5: Via printkey lässt sich der Inhalt eines einzelnen Schlüssels inspizieren

- Weitere Funktionalität umfasst die Auflistung aller User-Assist-Schlüssel [6.3.1.2.2] via **userassist**. Mit dem Befehl **hivedump** werden alle Keys in einem Hive rekursiv aufgelistet. Es kann auch sinnvoll und nötig sein, Anmeldedaten eines Benutzers für eine Domäne aus der Registry zu extrahieren - hierfür stellt Volatility den Befehl **hashdump** zur Verfügung. Zugehörige Passwörter können bei Bedarf entschlüsselt werden.

Es ist wichtig, zur Kenntnis zu nehmen, dass man in der Praxis eine Vielzahl von Ausprägungen von schadhaftem Code findet. Eine präzise Aussage, an welchen Stellen Artefakte in der Registrierungsdatenbank hinterlassen werden, ist daher nicht möglich.

6.3.1.1.5. Fortgeschrittene Malware-Analyse

Liefert die Speicheranalyse keine aufschlussreichen Ergebnisse oder sollen weitere Spuren sichergestellt werden, kann eine fortgeschrittene Malware-Analyse neue Anhaltspunkte liefern.

Schadhafte Software ist mitunter sehr komplex. Im Folgenden geht es deshalb keineswegs darum, alle möglichen Aspekte der Speicheranalyse zu beleuchten. Vielmehr sollen nur einige ausgewählte konzeptionelle Grundlagen erläutert werden, die rasche Lösungsmöglichkeiten in Aussicht stellen. Eine gute Einführung bietet beispielsweise [Hala].

Analyse von schadhafter Software im Speicher lässt sich nach einigen Kriterien einteilen. Die Beantwortung dieser Fragestellungen ist oft sehr schwierig und muss häufig auf Basis von Annahmen, Erfahrungen und Schätzungen erfolgen. Die Antworten auf diese Fragen hängen auch auf komplexe Weise voneinander ab.

- Allgemeine Analyse: Der Arbeitsspeicher wird nach verdächtigen Artefakten gescannt (via malfind).
- Erkennung von API-Hooking: Das API-Hooking bezeichnet ein Verfahren, mit dem fremder Code in vertrauenswürdige Anwendung integriert wird, um deren Funktionsweise zu modifizieren oder um schadhaften Code zu tarnen - immerhin untersuchen Firewalls Datenpakete nach bestimmten Mustern und können diesen Tarnmechanismus oft nicht erkennen [Crea]. Volatility integriert die Funktion apihooks, die sowohl Kernel- als auch Benutzer-Modus-API-Hooks erkennen kann.
- Während der laufenden Untersuchung kann weiterhin mit Volatility ein Prozessspeicherabbild mit dem Befehl *procexedump* erstellt werden. Der Einsatz eines aktuellen Virenscanners zum Schutz der Analyseumgebung ist dabei unerlässlich. Nach der Extraktion kann eine umfassende String-Analyse vorgenommen werden. Darüber hinaus ist eine Prüfung der binären Datei mittels eines Online-Virenscanners (z.B. *virustotal.com*) denkbar. Für die Sicherstellung der Beweismittelkette ist durch die Berechnung diverser Prüfsummen gesorgt.

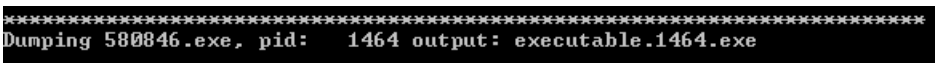

Abbildung 6.9.: procmemdump extrahiert Prozessspeicherabbild.

- Neben den beiden bereits genannten Kennfeldern gibt es weitere Verfahrensweisen zur Erkennung von Malware, die Auswirkungen auf die weitere Vorgehen haben könnten. Sie werden nur Vollständigkeit halber erwähnt. Sie sollten erst dann zum Einsatz kommen, wenn die vorhin genannten Verfahrensweisen keine Ergebnisse gebracht haben. Zum Beispiel können sog. YARA-Regeln verwendet werden, die das Schadsoftware anhand von charakteristischen Strings und Byte-Folgen identifizieren können. Hierzu ist es allerdings erforderlich, eigene Signaturen zu pflegen. Alternativ kann mittels YARA auch eine einfache String-Suche im Hauptspeicherabbild ausgeführt werden.

Nach erfolgreicher Kompromittierung eines Systems und Erreichen der Administrations-Ebene verwenden Kernel-Rootkits meistens einen Treiber, um sich in einem System unbemerkt einzunisten - sie können so das Verhalten des Betriebssystems beeinflussen und erzwingen, dass sie mit herkömmlichen Mitteln nicht entdeckt werden [Gil]. Erst durch eine detaillierte Betrachtung der installierten Treiber mittels Arbeitsspeicheranalyse können sie enttarnt werden. In Volatility beruht dieser Prozess auf der Funktion *devicetree*.

Eine weitere praktische Verfahrensweise, die momentan nur in Volatility integriert ist, ist die Überprüfung der Kernel Callback-Funktionen. Viele Windows-Kernel-Bestandteile, Antivirus-Programme und Rootkits nutzen sie, um Ereignisse zu beobachten und auf Veränderungen des Systemzustands reagieren zu können.

Weitere Informationen zu den hier vorgestellten Verfahren sowie noch einige weitere Funktionen des Volatility Frameworks sind unter [Hala] zu finden.

6.3.1.2. Analyse der Registrierungsdatenbank

Gründe für die Initiierung: Einschätzung des Schadens, vom Angreifer hinterlassene Spuren finden, zeitlichen Verlauf des Angriffs rekonstruieren.

Verantwortlich für Umsetzung: IT-Forensiker

Geschätzter Aufwand: mittel

Windows Server 2008R2 verwendet, wie alle Microsoft Betriebssysteme nach Windows 3.1, eine umfangreichere zentrale Datenbank, die hauptsächlich die Konfiguration der installierten Hard- und Software sowie User-(spezifischen)-Einstellungen enthält - die Registry. Diese Informationen werden bei jedem Start bei Bedarf in den Arbeitsspeicher eingelesen. Die Registry sichert kontinuierlich Daten, um die Funktionalität des Betriebssystems sicherzustellen. Veränderung, die der Nutzer vornimmt, werden automatisch in die Registrierung eingetragen. Das gleiche gilt auch für jede neu installierte oder entfernte Hard- und Software - damit ist die Registrierungsdatenbank eine wichtige Quelle für forensische Spuren. Dabei spielen folgende Fragen eine maßgebliche Rolle:

- Welche Nutzernamen gibt es und wann wurden diese zuletzt verwendet?
- Welche Anwendungen wurden zuletzt installiert bzw. genutzt?
- Welche Hardwaretreiber bzw. Dienste sind installiert?
- Auf welche Dateien wurde zuletzt zugegriffen?
- Welche Kommandos wurden zuletzt in der Kommandozeilenumgebung (cmd.exe) aufgerufen?
- Zeitstempel-Analyse: welche Schlüssel wurden zuletzt angelegt bzw. verändert? Gibt es auffällige Zeitstempel?

Nachfolgend wird Aufbau und Inhalte der Registry vorgestellt und mögliche Lösungsansätze für forensische Ermittlungen vorgestellt.

6.3.1.2.1. Aufbau der Registry

Die Registry besteht sowohl aus flüchtigen als auch nichtflüchtigen Daten und ist in einer Baumstruktur (ähnlich dem Windows-Explorer) aufgebaut. Sie besteht aus fünf Hauptschlüsseln, sog. Hives, die mit HKEY_ (engl. für „handle to key“) beginnen. Die Daten der Registry werden in mehreren Dateien in einem speziellen binären Format auf der Festplatte gespeichert. Ihr Pfad ist fest vorgegeben. Tabelle 6.1 listet alle Hives, ihr Speicherort und gibt Aufschluss darüber, ob die Daten volatil sind.

Schlüssel	Dateipfad
HKEY_LOCAL_MACHINE\System	%SystemRoot%\system32\config\System
HKEY_LOCAL_MACHINE\SAM	%SystemRoot%\system32\config\Sam
HKEY_LOCAL_MACHINE\Security	%SystemRoot%\system32\config\Security
HKEY_LOCAL_MACHINE\Software	%SystemRoot%\system32\config\Software
HKEY_USERS\Default	%SystemRoot%\system32\config\Default
HKEY_USERS\%User SID%	\Users\%UserProfile%\Ntuser.dat
HKEY_USERS\%User SID%_Classes	\Users\%UserProfile%\AppData\Local\Microsoft\Windows\Usrclass.dat
HKEY_LOCAL_MACHINE\Hardware	Volatiles Hive
HKEY_LOCAL_MACHINE\System\Clone	Volatiles Hive

Tabelle 6.1.: Speicherorte der Registry-Hives auf der Festplatte

Zwei der in der Tabelle letztgenannten Schlüssel existieren nur im Arbeitsspeicher [Car09].

Im Rahmen von Registry-Analyse sind grundsätzlich immer folgende Datentypen zu unterscheiden:

Schlüssel	Dateipfad
REG_BINARY	Binärer Wert
REG_DWORD	Integer-Wert, typischerweise 32-Bit lang
REG_SZ	Zeichenkette im UTF-16LE-Format, wird typischerweise durch ein Nullzeichen terminiert
REG_EXPAND_SZ	Zeichenkette im UTF-16LE-Format, variable Länge, kann Umgebungsvariablen enthalten, wird typischerweise durch ein Nullzeichen terminiert
REG_MULTI_SZ	Eine geordnete, durch Komma getrennte Liste von Zeichenketten im UTF-16LE-Format, ein Textstring wird durch ein Nullzeichen terminiert, die Liste durch ein zus. Nullzeichen
REG_NONE	kein Datentyp
REG_QWORD	Integer-Wert, typischerweise 64-Bit lang
REG_LINK	symbolischer Link im Unicode-Format, Verweis auf einen anderen Registry-Schlüssel
REG_RESOURCE_LIST	verschachteltes Array zur Speicherung einer Ressourcenliste (wird von der Plug-n-Play-Funktionalität verwendet)
REG_RESOURCE_REQUIREMENTS_LIST	verschachteltes Array zur Speicherung von Treiber-Listen (wird von der Plug-n-Play-Funktionalität verwendet)
REG_FULL_RESOURCE_DESCRIPTOR	verschachteltes Array zur Speicherung eines Ressourcen-Deskriptors (wird von der Plug-n-Play-Funktionalität verwendet)

Tabelle 6.2.: Verschiedene Datentypen der Registrierungsdatenbank im Überblick [Wikh]

6.3.1.2.2. Analyse der Registry

Im Rahmen von diesem Kapitel wird festgelegt, welche Schlüssel beim Vorliegen eines Security-Incidents am LRZ überprüft werden sollten. Solche Schlüssel sind auf der Basis von diversen Vorarbeiten bekannter Forensik-Experten ausgewählt worden - beispielsweise hat Harlan Carvey eine Excel-Liste im Rahmen seines Buches „Windows Forensic Analysis DVD Toolkit“ mit solchen Schlüsseln veröffentlicht [Car09].

Weiterhin hängt die Analyse der Registrierungsdatenbank wesentlich davon ab, wie versiert Angreifer vorgegangen sind - sie kombinieren und verändern Speicherorte der Informationen, um Schadcode unbemerkt in der

Registry einzunisten. Es gibt leider keine „out-of-the-box“ Lösung, die sich auf alle Sicherheitsvorfälle anwenden lässt - immerhin können Informationen sehr schnell ihre Gültigkeit verlieren. Eine umfassende Kenntnis der Struktur der Registry ist daher als Grundlage unabdingbar. Trotzdem kann dieses Kapitel wichtige Anhaltspunkte liefern, wo man nach Spuren suchen könnte und bildet insofern einen weiteren Ausgangspunkt für forensische Maßnahmen.

Im Bereich der forensischen Registry-Analyse spielen die folgenden Aspekte eine zentrale Rolle (die durchaus auch kombiniert zum Einsatz gelangen können):

- Überprüfen der nützlichen Schlüssel,
- Analysieren von Schlüssel, die von verdächtigen Prozessen geöffnet wurden.

Im Rahmen einer gründlichen forensischen Registry-Analyse spielen eine Reihe von Schlüsseln eine zentrale Rolle. Weiterhin muss berücksichtigt werden, dass bei Windows Server 2008R2 die Registrierungsdatenbank in 32-Bit- und 64-Bit-Bereiche unterteilt ist. 32-Bit-Schlüssel werden in einem separaten Bereich abgelegt, der unter dem folgenden Registrierungsschlüssel zu finden ist: *HKEY_LOCAL_MACHINE\Software\WOW6432Node* [Liga].

Wenn der Verdacht besteht, dass Malware im Spiel ist, gilt es, die Situation genau zu analysieren. Unter Windows Server 2008 R2 gibt es mehrere Möglichkeiten Software automatisch über die Registrierungsdatenbank nach einem Boot-Vorgang zu starten. Fast alle aktuellen softwarebasierten Schädlinge schreiben sich in einen der folgenden Schlüssel, um den Zugriff auf das System sicherzustellen.

So ist beispielsweise mögliche Modifikation der Schlüssel zu nennen, die für den User Logon Prozess relevant sind - unter Logon versteht man Elemente, die nach einer erfolgreichen Benutzeranmeldung ausgeführt werden:

- *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*
- *HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce*
- *HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run*
- *HKCU\Software\Microsoft\Windows\CurrentVersion\Run*
- *HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce*
- *HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnceEx*
- *HKLM\SOFTWARE\Microsoft\Active Setup\Installed Components*
- *HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\Credential Providers*

Zwingend nötig für die Registry-Analyse ist weiterhin Überprüfung der registrierten Dienste sowie Treiber, da oft auf diesem Wege schadhafte Software in ein System eingeschleust und daraufhin unbemerkt ausgeführt wird - eine Benutzeranmeldung ist nicht für die Ausführung erforderlich.

- *HKLM\System\CurrentControlSet\Services*

Überprüfen von weiteren möglichen Autostarteinträgen spielt für viele Betrachtungen eine ganz zentrale Rolle. Folgende drei Schlüssel enthalten Programmeinträge, die beim Systemstart ausgeführt werden sollen:

- *HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run*
- *HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run*
- *HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\SharedTaskScheduler*

Benutzer führen viele verschiedene grundlegende Funktionen - beispielweise Öffnen von Dateien oder Programmen - auf einem System aus. Eine präzise Überprüfung ist ratsam, falls ungewöhnliche Systemaktivitäten bei bestimmten Aktionen, z.B. Starten von Applikationen beobachtet wurden. Eine Vielzahl von Einträgen kann im *HKLM\SOFTWARE\Classes*-Hive gefunden werden, *exefile* wird beispielsweise beim Start von ausführbaren Dateien ausgewertet - weicht der Wert von *“%1“ %** ab, so ist dies ein Indiz für eine Modifikation durch schadhafte Software.

Es darf nicht außer Acht gelassen werden, dass in einigen Fällen Windows-Firewall umkonfiguriert werden kann [Abbildung 6.10]. Damit lässt sich schadhafter Netzverkehr an der Firewall vorbei ins Netz schleusen.

- *HKLM\SYSTEM\CurrentControlSet\services\SharedAccess\Parameters\FirewallPolicy\StandardProfile*

In einigen Fällen - wenn beispielsweise das System sofort nach dem Bekanntwerden eines Security Incidents heruntergefahren wurde und keine Live-Response-Daten vorliegen - sollten zuletzt benutzte Dateien, zuletzt benutzte Programme sowie zuletzt eingegebene Befehle in der CLI-Umgebung überprüft werden. Entsprechende Registry-Schlüssel sind:

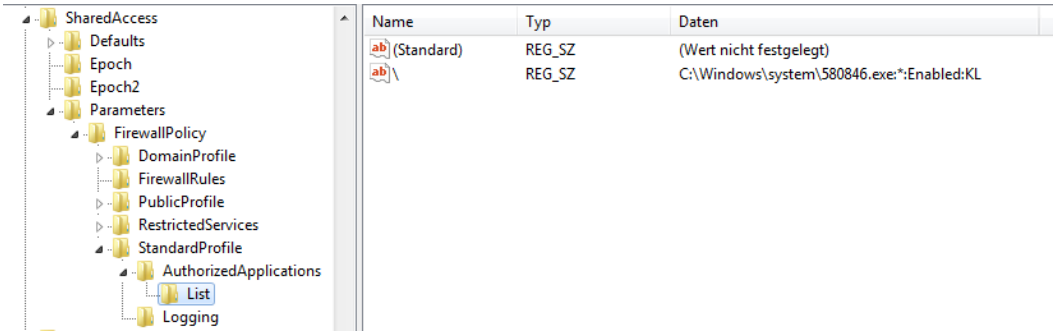

Abbildung 6.10.: Schadcode modifiziert Firewall-Regeln, um Kommunikation mit dem C&C-Server sicherzustellen.

- *HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU*
- *HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedMRU*
- *HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs*
- *HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU*
- *HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\GUID\Count*

Es existieren noch zahlreiche weitere Stellen in der Registrierungsdatenbank, die durch Malware-Autoren missbraucht werden können (und es kommen immer wieder neue Varianten hinzu) - hier zeigt sich, dass die detaillierte und sorgfältige Analyse der Registry eine langfristige Herausforderung für das Leibniz-Rechenzentrum ist und kaum manuell zu bewerkstelligen ist. Hinzu kommt, dass Werte manchmal entweder im Hexadezimalsystem vorliegen oder im ROT13-Format (auch bekannt unter dem Namen Caesar-Verschlüsselung) verschlüsselt sind - dabei werden einzelne Buchstaben um 13 Zeichen im Alphabet verschoben.

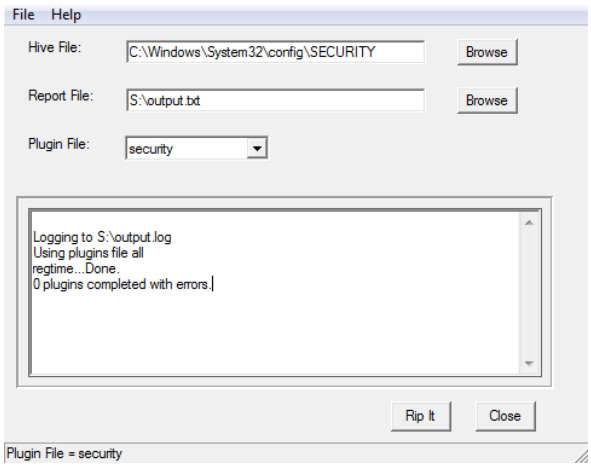

Abbildung 6.11.: RegRipper analysiert Registry-Keys unter forensischen Gesichtspunkten.

Aus forensischen Gesichtspunkten sollte weiterhin *Last Write Time* Berücksichtigung werden: ein 64-bittiger FILETIME Wert, welcher den Zeitpunkt des letzten Schreibzugriffs auf den aktuellen Schlüssel anzeigt und in etwa den MAC-Zeiten bei Dateisystemen entspricht. Sortiert man die Registry nach den Zugriffszeiten, so ist man in der Lage, eine Zeitlinie für die letzten Registry-Operationen aufzustellen.

Genau an dieser Stelle setzt RegRipper an - ein hervorragendes Framework von Harlan Carvey zur komfortablen Analyse diverser Registry-Keys. Voraussetzung für den Einsatz des Tools ist ein forensiches Duplikat der Registry-Hives. RegRipper wird mit diversen Plug-ins ausgeliefert und wird ständig um neue Funktionalitäten erweitert. Der Bericht über forensisch interessante Registry-Schlüssel wird in einer Textdatei umgeleitet und kann in Ruhe analysiert werden: ein besonderes Augenmerk liegt dabei auf der Analyse von Nutzungsspuren.

Sicherlich kann auch RegRipper nicht die komplette Registrierungsdatenbank analysieren. Es darf nicht außer Acht gelassen werden, dass Malware an verschiedensten Stellen Einträge anlegen kann - Tests zeigen aber, dass die Überprüfung der genannten Stellen ausreichend Hinweise liefern kann.

6.3.1.3. Logdatei-Analyse

Gründe für die Initiierung: Art des Angriffs ermitteln, Angriffsquelle & Schwachstelle bestimmen, Ausmaß des Schadens quantifizieren, Angriffszeitpunkt ermitteln, Korrelation der Daten.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: hoch

Auf einem Windows Server 2008R2 existiert bereits in der Standardkonfiguration eine Myriade von Logdateien. Die Untersuchung dieser großen Datenmenge ist zwar mühsam, aus forensischer Sicht allerdings notwendig, vor allem dann, wenn sich der Angreifer keine Mühe macht, seine Spuren zu verwischen und die relevanten Log-Einträge zu säubern. Im folgenden Abschnitt werden ausgewählte Protokolldateien und Methoden für die forensische Analyse eines Windows-Server-2008-Systems vorgestellt.

Ganz bewusst wird im Folgenden nur auf drei Arten von Protokolldateien eingegangen - immerhin steigt der Arbeitsaufwand im Verhältnis zu Datenmenge linear an. Außerdem würde eine ausführliche Behandlung aller Möglichkeiten den Rahmen dieses Leitfadens sprengen, nämlich:

- Ereignisprotokolle (Dazu zählen insbesondere das Anwendungs-, das System-, das Installation- und das Sicherheitsprotokoll)
- IIS-Protokolle,
- sowie Aufgabenplanungsprotokolle.

Idealerweise können nach dem Abschluss der Auswertung unter anderem folgende Fragestellungen beantwortet werden [Ges10]:

- Wurden Protokolldateien modifiziert oder komplett gelöscht?
- Existieren fehlgeschlagene Anmeldeversuche (besonders bei Usern mit erweiterten Rechten oder Administratoren)?
- Existieren erfolgreiche Anmeldungen im Vorfeld von einem SI?
- Wurden Aktionen ausgeführt, die erweiterte Rechte voraussetzen (z.B. Hinzufügen, Starten oder Stoppen von Diensten, Ändern der Systemkonfiguration, etc)?
- Wurde die Ausführung von Aktionen unterbunden, die privilegierte Rechte voraussetzen?
- Existieren neue Dienste, Programme oder Benutzerkonten?
- Wurden Benutzerzugriffsrechte modifiziert bzw. angehoben?
- Existieren versuchte bzw. erfolgreiche Zugriffe auf geschützte Dateien?
- Tauchen vermehrt Meldungen im Ereignisprotokoll auf?
- ...

Der Dreh- und Angelpunkt der Logdatei-Analyse ist die Datenreduktion. Eine Datenreduktion wird notwendig weil - vor allem auf stark frequentierten Systemen - die zu untersuchende Datenmenge sehr schnell anwächst je weiter der Vorfall zurückliegt. In erster Linie sollen dabei Daten entfernt werden, die für die Untersuchung irrelevant sind, d.h. zunächst Ereignisse, die als normal einzustufen sind. Das setzt natürlich voraus, dass die Ermittler mit dem Windows-System vertraut und in der Lage sind harmlose Einträge von Anomalien zu unterscheiden. Der Einsatz eines Log-Parsers (z.B. Microsoft LogParser [Supb] in Kombination mit Visual LogParser GUI [mch] oder LogParser Lizard [Lab]) erleichtert die Aufgabe zusätzlich. Die Verwendung geeigneter Werkzeuge ist umso wichtiger, da - wie bereits erwähnt - bei größeren Systemen die Datenmenge sehr umfangreich ausfallen kann.

6.3.1.3.1. Ereignisprotokolle

Der Ereignisprotokolldienst ähnelt sehr dem Syslog-Dienst auf Linux-basierten Systemen und generiert auf einem Windows Server 2008R2 System zuverlässig Ereignisse, die bei der Überprüfung von Sicherheitsoperationen, des Zugriffs auf Systeme u.v.m. nützlich sind. Diese werden in mindestens vier Protokolldateien, nämlich dem Anwendungs-, dem System-, dem Installation- und dem Sicherheitsprotokoll, je nach Quelle des Überwachungsereignisses, hinterlegt. Weiterhin können, abhängig von der Systemkonfiguration sowie Vorhandensein weiterer Anwendungen, zusätzliche Standardprotokolldateien zu finden sein.

Für forensische Untersuchungen eines Windows-Servers am LRZ werden tiefgreifende Kenntnisse über Ereignisprotokolle zur Datenreduktion vorausgesetzt. Eines der ersten Schritte ist eine Betrachtung der sog. Ereignis-ID (Event ID). Mittels Ereignis-ID werden bestimmte Ereignisse isoliert betrachtet. Die benötigten Informationen, zu denen eine Zuordnung der IDs zur spezifischen Ereignissen zählt, sind von Microsoft ausführlich dokumentiert [Micb]. Eine kleine Auswahl kann der Tabelle 6.3 entnommen werden:

Ereignis-ID	Beschreibung
4612	Verlust von Protokolldaten
4624	erfolgreiche Benutzeranmeldung
4625	erfolgloser Anmeldeversuch
4634	erfolgreiche Benutzerabmeldung
4649	Anmeldeversuch mit expliziten Anmeldeinformationen
4649	Feststellung eines Replay-Angriffs
4672	Zuweisung besonderer Rechte nach einer Anmeldung
4674	versuchte Benutzung von privilegierten Systemkommandos
4691	indirekter Zugriff auf ein Objekt wurde angefordert
4704	Erweiterung der Zugriffsrechte
4719	Änderung der Überwachungsrichtlinien
4720	Anlegen eines neuen Benutzerkontos
4722	Aktivieren eines Benutzerkontos
4723	Versuch der Kennwortänderung für ein Benutzerkonto
4723	Versuch der Kennwortrücksetzung für ein Benutzerkonto
4726	Löschung eines Benutzerkontos
4738	Änderung eines Benutzerkontos

Tabelle 6.3.: Beschreibung ausgewählter Sicherheitsereignisse und der ihnen zugeordneten IDs

6.3.1.3.2. IIS-Protokolle

Bei den Windows-Server-Systemen am Leibniz-Rechenzentrum kommt Internet Information Services (IIS) (vormals Internet Information Server) als Webserver zum Einsatz. Wie es bei Web-Komponenten fast schon zur Regel gehört, werden immer wieder neue Angriffsmöglichkeiten entdeckt und für einen Angriff auf ein IT-System ausgenutzt, die unter bestimmten Voraussetzungen vom erfolgreich sind. Oft lassen sich Angriffe

durch die Analyse der IIS-Protokolldateien nachweisen. Gute handwerkliche Fähigkeiten und die Kenntnis der IIS-Plattform wird vorausgesetzt.

Die IIS-Logs sind in aller Regel im *%SystemRoot%\system32\Logfiles* Verzeichnis zu finden, wobei für jeden Host ein eigenes Unterverzeichnis angelegt wird. Weiterhin gilt: typischerweise liegen die Protokolldateien im ASCII-Format, somit können sie in jedem beliebigen Text-Editor geöffnet werden. Um eine erfolgreiche Analyse in die Wege zu leiten, sollte zunächst eine Datenreduktion vorgenommen werden - dieser Schritt kann entfallen, falls bereits Informationen vorliegen, die auf eine bestimmte Vorgehensweise des Angreifers schließen lassen und eine einfache Stringsuche ausreichend ist.

Die Thematik der möglichen IIS-Angriffe ist sehr umfangreich - sie allumfassend zu bearbeiten würde den Rahmen dieser Seiten sprengen und ist auch nicht das Ziel. Es folgen lediglich einige Beispiele für mögliche Basisszenarien:

- Bei SQL-Injection-Angriffen maskieren Angreifer die SQL-Befehle durch Verwendung von Escape-Sequenzen, bestehend z. B. aus % und zwei hexadezimalen Ziffern, oder Verwendung der hexadezimalen Kodierung. An dieser Stelle kann man sich zu Nutze machen, dass solche Anfragen typischerweise viel länger sind, als die üblichen URIs (Universal Resource Identifier). Durch die Auswertung des cs-uri-query-Feldes können solche Angriffe schnell lokalisiert werden [Listing 6.6 [McD]].

```
orderitem.asp?IT=GM-204;DECLARE%20@S%20NVARCHAR(4000);SET%20@S=CAST(0x44004500430
04C004100520045002000400054002000760061007200630068006100720028003200350035002900
2C004000  43002000760061007200630068006100720028003200350035002900200044004500430
04C004100520045002000540061006200 6C0065005F00430075007200730006F00720020004300550 0
520053004F00520020004600 4F00520020007300650 06C00650063007400200061002E006E0061006
D0065002C0062002E006E0061006D00650020006600 72006F006D00200073007900730073006F0062006A
00650063007400730020006100 2C00730079007300630 06F006C0075006D006E00730020006200200
07700680065007200650020006100 2E00690064003D0062002E00690064002000610 06E0064002000
61002E00780074007900700065003D00270075002700200061006E00640020002800620 02E0078007
40079007000650 03D00390039002000 6F00720020006200 2E00780074007900700065003D00330035
```

Listing 6.6: Beispiel für einen SQL-Injection-Angriff

- Mit Hilfe von überlangen, in schneller Abfolge erfolgten URI-Anfragen können Angreifer eine vollständige Auslastung eines Windows-Servers verursachen, wodurch die IIS-Komponente keine Anfragen mehr beantworten kann. Die Vorgehensweise bei der Logfile-Analyse entspricht dem ersten Beispiel.

Bei bekannten Schwachstellen kann es sich als äußerst vorteilhaft erweisen, falls bisher gesammelten Erkenntnisse sich mit Informationen über Schwachstellen korrelieren lassen, wie sie beispielsweise von DFN-CERT mitgeteilt werden. Eine weitere Anlaufstelle stellt die Seite „Microsoft Security Bulletins“ [Micc] dar. Dadurch lässt sich die Auswertung auf bestimmte Einträge eingrenzen.

Weiterhin sollte in Erwägung gezogen werden, basierend auf erkannten Angriffen, eine Liste von Signaturen durch das LRZ-CSIRT für einfache Stringsuchen zu pflegen. Solche Signaturen könnten wiederkehrende Schlüsselbegriffe enthalten, wie z.B. DECLARE, CAST, NVARCHAR, sowie auch Informationen zu den möglichen Schwachstellen und aktuellen Bedrohungen integrieren.

6.3.1.3.3. Aufgabenplanung-Protokolle

Windows Server 2008R2 enthält einen Aufgabenplanungsdienst zur Automatisierung von Programmen und Skripten. Er wird typischerweise von Administratoren für Routineaufgaben benutzt, kann aber auch von Malware (z.B. Conflicker [Supc]) verwendet werden, um sicherzustellen, dass der schadhafte Code in regelmäßigen Abständen, z.B. beim Eintreffen von einem bestimmten Ereignis oder Systemstatusänderungen ausgeführt werden kann.

Windows Server-Aufgabenplanung führt ausführliche Protokolle, die für forensische Analyse verwendet werden können. In der Standardkonfiguration kann die entsprechende Protokolldatei (*SCHEDLGU.TXT*) im Verzeichnis *%SystemRoot%\Tasks* gefunden werden, der aktuelle Pfad befindet sich im Registry-Schlüssel *HKLM\Software\Microsoft\SchedulingAgent*.

Leider bietet die Protokolldatei keine Information zum Speicherort der ausgeführten Datei, liefert aber zumindest einen Anhaltspunkt, wann ein Programm gestartet wurde. Für eine effektive Ermittlung ist es sinnvoll,

alle zur Verfügung stehende Informationen zum zeitlichen Ablauf zu sammeln. Dies sorgt für eine stichhaltige Korrelation der Ergebnisse.

6.3.1.4. Erkennung von Rootkits

Geschätzter Aufwand: gering

Gründe für die Initiierung: Verdacht auf Malware

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Wie in jedem großen IT-Betrieb hat das LRZ-Personal häufig mit Kompromittierungen auf Basis von Rootkits zu tun. Die Folgen einer Infektion können sehr weitreichend sein und u.a. zum Fehlverhalten der betroffenen Systeme führen - immerhin werden oft interne Abläufe des Betriebssystems abgeändert. Bei einer genauen Betrachtung von Windows-Rootkits wird deutlich, dass sie sich in zwei Bereiche, entsprechend ihrer Privileg-Stufen, einteilen lassen: Benutzer-Modus- sowie Kernel-Modus-Rootkits [Wikg]. Nachfolgend werden die jeweils zu berücksichtigenden Aspekte bei der Bekämpfung von Rootkits dargestellt.

Folgende Vorgehensweisen sind zu unterscheiden:

- Erkennung während der Post-Mortem-Analyse,
- Erkennung am lebenden System.

Ganz grob soll ferner die Indikation für das Vorhandensein eines Rootkits angeschnitten werden. Unter anderem sind folgende Informationen für die Diagnose relevant:

- Unerwartete Neustarts des Systems,
- Fehlverhalten bei Ausführung von bestimmten Funktionen,
- Ungewöhnliche Fehlermeldungen,
- Änderung der Systemkonfiguration,
- Ungewöhnliches Kommunikationsverhalten.

6.3.1.4.1. Post-Mortem-Analyse

Die genaue Vorgehensweise ist abhängig von den zur Verfügung stehenden Daten und ist mit dem SIC abzustimmen. Zu den möglichen Methoden zählen:

- Erkennung von Rootkits mittels Hauptspeicher-Analyse. Es ist die zuverlässigste, wenn auch teilweise umständliche, Technik. Die Vorgehensweise sollte sich am Abschnitt 6.3.1.1 orientieren. Falls kein Hauptspeicherabbild vorliegt kann eines der folgenden Varianten gewählt werden.
- Durchführung eines spezifischen Datei-Scans zur Erkennung von Rootkits. Hierfür wird das zu untersuchende Abbild schreibgeschützt in das Analysesystem eingebunden (siehe Abschnitt 6.2) und mit einem oder mehreren Anti-Malware-Lösungen untersucht. Bedingt durch die Verwendung des Schreibschutzes werden keine Änderungen an den Dateien vorgenommen und die Beweisintegrität bleibt erhalten. Weiterhin gilt: ein Rootkit bleibt inaktiv: damit werden Dateien, Verzeichnisse oder Registry-Schlüssel nicht gefiltert und die Tarnmechanismen greifen somit nicht.
- Analyse der Registry. Die dritte Möglichkeit für die Rootkit-Diagnose bezieht sich auf die Analyse der *Services* Registry-Schlüssel und die Sortierung nach LastWrite Time. Wichtig dabei ist alle Einträge zu betrachten, es existiert in aller Regel in jedem ControlSet ein Services-Key. Mit dem bereits vorgestellten Tool RegRipper [6.3.1.2.2] lässt sich dieser Vorgang vereinfachen. Das Programm liest alle Schlüssel, wie etwa *HKLM\System\CurrentControlSet\Services* ein und sortiert sie nach ihren Zugriffszeiten.

6.3.1.4.2. Live-Analyse

Bei einer Rootkit-Analyse eines lebenden Systems lassen sich grundsätzlich zwei mögliche Vorgehensweisen identifizieren:

- Erkennung von Rootkits mittels Datenabgleich. Die Idee dahinter ist simpel - Daten aus zwei verschiedenen Quellen werden erfasst und miteinander verglichen. Oft lassen sich zur Laufzeit eines Systems dadurch Auffälligkeiten bzw. Diskrepanzen erkennen, die auf schadhafte Software hindeuten. Ein Beispiel wäre ein lokaler und ein externer Port-Scan - an dieser Stelle sei auf den Abschnitt 5.6.2.4 verwiesen.
- Durchführung eines spezifischen Datei-Scans zur Erkennung von Rootkits. Die Analyse eines noch aktiven, nicht ausgeschalteten Systems fällt streng genommen nicht in den Bereich der Computer-Forensik - es werden viele Artefakte auf dem Host hinterlassen und die Integrität der Beweismittel kann nicht garantiert werden. Dieser Ansatz ist sinnvoll, wenn das System aus Verfügbarkeitsgründen nicht ausgeschaltet werden kann oder aber eine forensische Untersuchung nicht im Vordergrund steht. Es existieren eine Reihe von Tools, die Rootkits aufspüren können: F-Secure BlackLight [FS], Sophos Anti-Rootkit [Sop], RootkitRevealer [Rusb] sowie GMER [GME], nur um einige zu nennen. Abhängig von der Komplexität der verwendeten Tarntechniken kann eine solche Untersuchung ergebnislos bleiben - vor allem die Präsenz der Kernel-Modus-Rootkits kann zum Teil, bedingt durch das Anpassen des Betriebssystemkerns, unerkannt bleiben.

Unabhängig davon welche Vorgehensweise gewählt wird, sollte die Partitionierung der angeschlossenen Datenträger ebenfalls betrachtet werden. Der freie Speicherplatz kann für den Ermittlungsverlauf wesentliche Informationen enthalten. Einige Rootkits verbergen sich in einer versteckten Partition. Der neue, unter Windows Server 2008R2 lauffähige Rootkit TDL4 erstellt zum Beispiel eine kleine Partition im nicht alloziierten Bereich eines Datenträgers und lagert den Schadcode dorthin aus - dabei wird der Umstand ausgenutzt, dass bei Windows Server ein kleiner Festplattenbereich meistens unpartitioniert bleibt. Anschließend wird die Partitionstabelle und somit der Bootvorgang modifiziert, damit die Rootkit-Komponenten vor dem Start des Betriebssystems in den Speicher geladen werden [Harb].

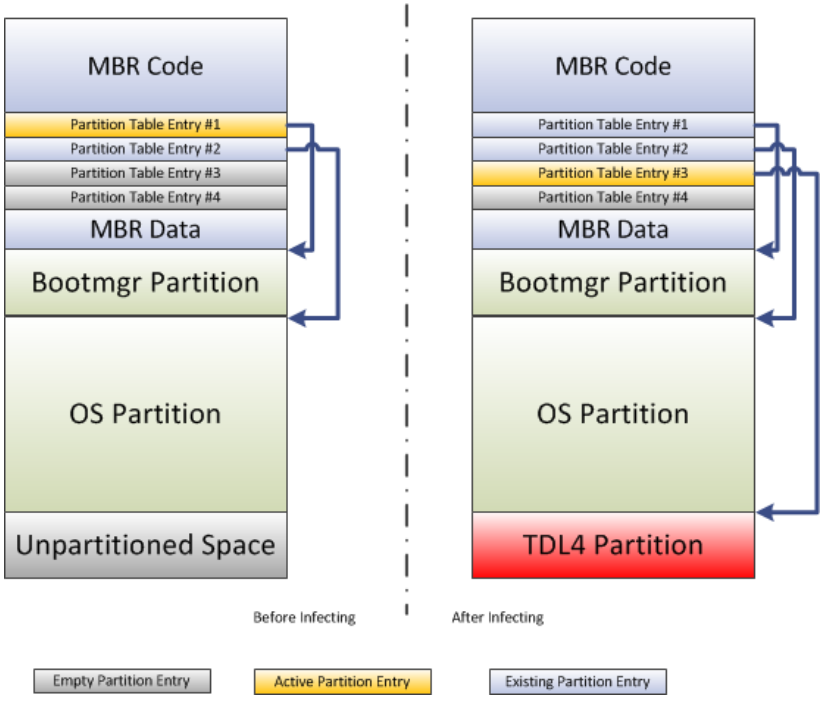

Abbildung 6.12.: TDL4-Rootkit nurtzt den nicht partitionierten Bereich einer Festplatte, um seine Dateien zu verstecken.

Quelle: [Harb]

6.3.1.5. Analyse des Dateisystems

Um Sicherheitsvorfälle vollkommen aufklären zu können, ist es manchmal notwendig, das Windows-Dateisystem zu analysieren. Typische Maßnahmen können die Wiederherstellung von gelöschten Daten, Analyse des freien Speicherplatzes oder Mac-Time-Analyse sein. Um die Beweisintegrität zu wahren, müssen alle im Folgenden vorgestellten Verfahren nur im Verbund mit einem forensischen Duplikat verwendet werden.

Der Fokus dieses Kapitels liegt auf der Einführung in den Bereich der Diskforensik. Dabei gilt es den LRZ-spezifischen Anforderungen Rechnung zu tragen: Effizienz und Minimierung der Bearbeitungszeit stehen für das LRZ an erster Stelle - daher sollten diese Maßnahmen, obwohl sie erprobt sind, nur dann angewendet werden, wenn der SIC den benötigten Zeitaufwand für notwendig erachtet. Steht die straf- bzw. zivilrechtliche Verfolgung nicht im Vordergrund, sollten die vorherigen Schritte genügend Anhaltspunkte geliefert haben, um ein SI abschließen zu können.

6.3.1.5.1. Windows Papierkorb

Gründe für die Initiierung: Zurückgelassene Spuren auf den Datenträgern analysieren, Hinweise auf gelöschte Dateien finden.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: gering

Wird unter Windows Server 2008R2 eine Datei zum Löschen markiert, wird sie in den Papierkorb verschoben und zwar in den Unterordner *C:\$Recycle.Bin\%User-SID%*. Weiterhin wird die Datei, entsprechend der Windows-Richtlinien, umbenannt: der neue Dateiname beginnt mit dem *$*-Zeichen und dem Buchstaben *R*, gefolgt von sechs zufälligen Buchstaben bzw. Zahlen sowie der ursprünglichen Dateierweiterung. Zeitgleich wird eine weitere Datei angelegt, deren Name mit *$*-Zeichen und dem Buchstaben *I* anfängt und ansonsten mit der Parent-Datei übereinstimmt. Folgende Informationen sind in dieser Datei enthalten:

- ursprünglicher Datei-Name und Speicherort,
- ursprüngliche Dateigröße,
- Lösch-Datum und -Uhrzeit.

Wenn der Papierkorb geleert wird, werden beide Dateien gelöscht [Mac]. Ein Beispiel kann der Abbildung 6.15 entnommen werden.

Um die Dateigröße auszulesen[3] , muss die Bytefolge 0x08-0x0F betrachtet werden. Die Dateigröße liegt lediglich im hexadezimalen Format vor und muss vor der Umrechnung in das Dezimalsystem invertiert werten [Abb. 6.13].

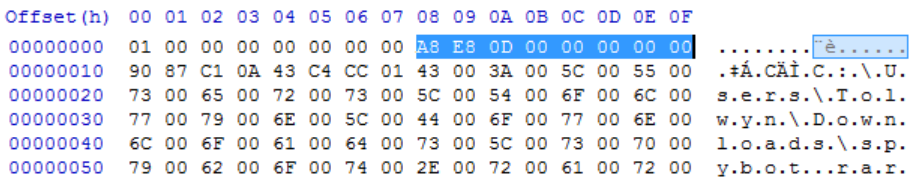

Abbildung 6.13.: Ursprüngliche Dateigröße kann mit Hilfe eines HEX-Editors ermitterlt werden.

Ähnlich ist auch beim Lösch-Datum und -Uhrzeit zu verfahren: es wird die Bytefolge 0x10-0x17 betrachtet - es handelt sich hierbei um einen Wert, der die Anzahl der 100-Nanosekunden-Intervalle darstellt, die seit dem 1. Januar 1601 vergangen sind. Dieser Wert muss für die Umrechnung ebenfalls invertiert werden. Typischerweise wird die dezimale Zahl anschließend in das Unixzeit-Format und mit Hilfe eines Converters in eine lesbare Darstellung umgewandelt. Hierfür eignet sich folgende Formel:

[3]Diese Informationen lassen sich auch mit Hilfe des FTK Imagers anzeigen - zur besseren Verständnis der Grundfunktionsweise wird die manuelle Auswertung erklärt.

$$\frac{Zeitstempel}{10000000} - 11644473600$$

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00000000	01	00	00	00	00	00	00	00	A8	E8	0D	00	00	00	00	00
00000010	90	87	C1	0A	43	C4	CC	01	43	00	3A	00	5C	00	55	00
00000020	73	00	65	00	72	00	73	00	5C	00	54	00	6F	00	6C	00
00000030	77	00	79	00	6E	00	5C	00	44	00	6F	00	77	00	6E	00
00000040	6C	00	6F	00	61	00	64	00	73	00	5C	00	73	00	70	00
00000050	79	00	62	00	6F	00	74	00	2E	00	72	00	61	00	72	00

Abbildung 6.14.: Lösch-Datum und -Uhrzeit einer Datei kann mit Hilfe eines HEX-Editors ermitterlt werden.

Ein weiterer Untersuchungsansatz beschäftigt sich mit Dateien im Hauptverzeichnis des Papierkorbs. Hierbei gelten alle Dateien als verdächtig, die sich nicht in den *%User-SID%*-Unterveichnissen befinden sowie Dateien, die nicht der Namenskonvention entsprechen. Allerdings sollte man sich auch bewusst sein, dass manche reguläre Applikationen Dateien im *C:\$Recycle.Bin* Verzeichnis ablegen können, zum Beispiel erstellt Norton's Recycle Bin Protector die Datei nprotect.log in diesem Ordner.

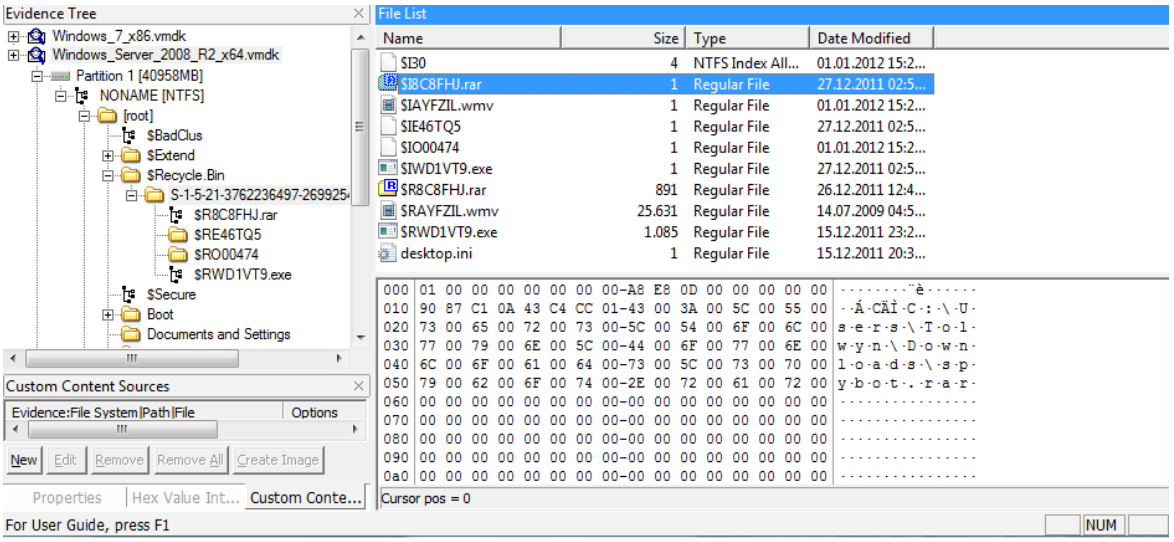

Abbildung 6.15.: Inhalt des Windows Server 2008 Papierkorbes im FTK Imager.

6.3.1.5.2. Wiederherstellung von gelöschten Dateien

Gründe für die Initiierung: Zurückgelassene Spuren auf den Datenträgern analysieren, Hinweise auf gelöschte Dateien finden.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: mittel bis hoch

Bei einem Einbruch werden vom Angreifer oft Dateien gelöscht und aus dem Papierkorb entfernt, um Spuren zu verwischen. Diesem Umstand muss Rechnung getragen werden, indem Verfahrensweisen zur Datenwiederherstellung bei Bedarf eingesetzt werden.

Oft lassen sich entweder Dateifragmente oder komplette Dateien von Datenträgern wiederherstellen - geeignete forensische Software, wie z.B. The Sleuth Kit, vorausgesetzt. So ein Vorgang kann mehrere Stunden dauern und es können hierbei viele Dateien anfallen.

Weiterhin sollte beachtet werden, dass die Grenzen der Datenwiederherstellung immer dann erreicht sind, wenn das Betriebssystem eine Datei nicht nur zum Löschen markiert, sondern deren Platz mit anderen Datenstrukturen belegt hat. Je mehr Zeit zwischen dem Eintreten eines Security Incidents und Wiederherstellungsversuch verstreicht, desto schwieriger gestaltet sich dieser Vorgang.

6.3.1.5.3. Versteckte Dateien

Gründe für die Initiierung: Zurückgelassene Spuren auf den Datenträgern analysieren, vom Angreifer maskierte Dateien finden.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: mittel

Das Dateisystem NTFS ermöglicht bereits seit seit Windows NT 3.51 das Speichern von Daten in alternativen Datenströmen (engl. Alternate Data Streams bzw. ADS). Das Betriebssystem kann zusätzliche Informationen an vorhandene Daten anbinden, um neben dem eigentlichen Dateiinhalt noch weitere sogenannte Streams in einer Datei abzuspeichern - ein Beispiel wäre das mit Windows XP SP 2 eingeführte Zone.Identifier [Supa], welches Dateien, die aus dem Internet stammen, kennzeichnet. Viele Windows-Programme nutzen solche Streams zur Speicherung von Vorschaubildern sowie zur Speicherung der Metadaten. Zu jeder Datei können beliebig viele Unter-Streams gespeichert werden.

Die Erfahrung zeigt, dass Malware-Autoren stellenweise die Funktionalität der ADS für ihre Angriffe ausnutzen und den schädlichen Code verstecken, denn nicht alle Virenscanner erkennen Malware in Streams zuverlässig. Um ADS anzeigen zu lassen, kann man auf den Befehl *dir* mit dem Schalter */r* zurückgreifen. Alternativ lässt sich auch spezialisierte Software nutzen, um ein forensisches Abbild auf das Vorhandensein von Alternate Data Streams zu überprüfen - ein Beispiel wäre LADS (List Alternate Data Streams) [Hey] oder ADS Spy [Mer].

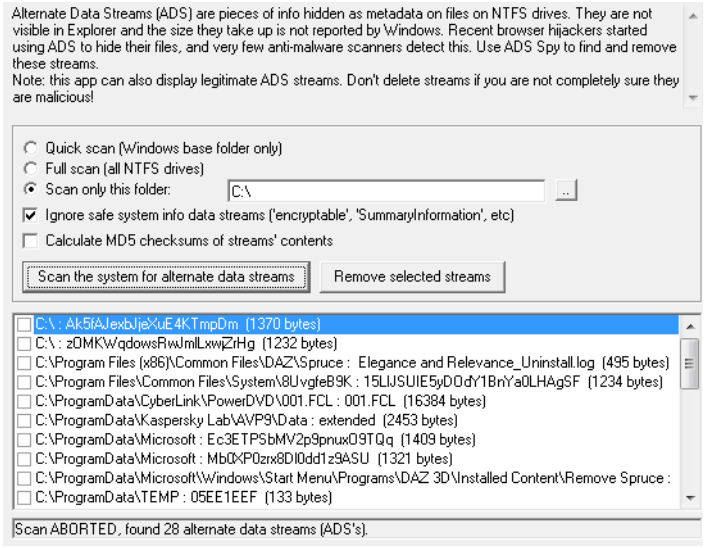

Abbildung 6.16.: ADS Spy findet ADS und listet sie übersichtlich auf.

6.3.1.5.4. Timeline Analyse

Gründe für die Initiierung: Einschätzung des Schadens, zeitlichen Verlauf des Angriffs rekonstruieren, Hinweise auf weitere Beweisfundorte finden.

Verantwortlich für Umsetzung: System-Administratoren, Netz-Administratoren

Geschätzter Aufwand: hoch

Die Timeline Analyse ist ein Mittel, das das LRZ-CSIRT-Team zur Identifizierung oder Verknüpfung einer Abfolge von Ereignissen verwenden kann. Sie umfasst in erster Linie Erstellung von aussagekräftigen Dateilisten, um die Dateien zu bestimmen, die während des Vorfalls geöffnet, aufgerufen oder sonst wie betrachtet wurden.

Meistens beginnt eine Entwicklung der Zeitlinie durch das Sammeln von zeitbezogenen Daten und das Einfügen in ein Exceldokument. Einer der Vorteile dieser Art der Entwicklung der Timeline ist, dass das Hinzufügen neuer entdeckter Ereignisse relativ einfach ist und sobald neue Daten hinzugefügt wurden, die Einträge sortiert werden können, um alle Einträge in die richtige Reihenfolge auf der Zeitlinie zu bringen. Dieser Prozess kann sehr zeitaufwändig und umständlich sein. Ferner ist dieses Verfahren nicht besonders skalierbar, da mit modernen Betriebssystemen und anderen Datenquellen der mit der Analyse betrauter Forensiker schnell durch die schiere Anzahl der Ereignisse, die eventuell Einfluss oder Relevanz für die Prüfung haben könnte, überwältigt wird.

Eine automatisierte Möglichkeit für die Sammlung von Zeitstempelinformationen aus einem erzeugten Image ist die Verwendung des *fls*-Tools - ein Bestandteil des The Sleuth Kits [Carb]. Das fls-Tool kann entweder direkt im Verbund mit einem forensischen Duplikat genutzt oder auf ein gemountetes Abbild angewendet werden - dabei werden kürzlich gelöschte Dateien und Verzeichnisse aufgelistet und in einem sog. „body-file“ gespeichert. Mit geeigneter Software lassen sich die Zeitangaben, die im Windows-Zeitformat (siehe Abschnitt 6.3.1.5.1) vorliegen, in ein verständlicheres Format konvertieren. Nähere Informationen hierzu sind bei [Carc] erhältlich.

6.3.1.6. Analyse von unbekannten Binärdateien

Gründe für die Initiierung: Verdacht auf schadhaften Code, Angriffsvektor identifizieren.

Verantwortlich für Umsetzung: System-Administratoren

Geschätzter Aufwand: hoch

Oft kann ein Eindringling Skripte oder Konfigurationen verändern, wobei es sich bei diesen Dateien in der Regel um Textdateien handelt, die geöffnet und angezeigt werden können. Gleichfalls wird es manchmal während der Untersuchung notwendig sein, die Art der ausführbaren Datei zu identifizieren, insbesondere wenn die verwendete Antiviruslösung noch nicht in der Lage ist, das Schadprogramm zu erkennen. Auf Dauer gesehen kann die manuelle Analyse der unbekannten Binärdateien möglicherweise zu einer schnelleren Reaktion auf ein Security Incident führen, da die Analyse eines Schadprogramms zu einem besseren Verständnis darüber führt wie es eindringen konnte, wie es auf Systemen am Laufen blieb und welche Artefakte es hinterlassen hat. Diese Artefakte können außerdem dann dazu benutzt werden, andere infizierte Systeme zu lokalisieren. Diese Maßnahme entspricht zum Teil der Methode „Fortgeschrittene Malware-Analyse“ (6.3.1.1.5] und sollte als komplementäre Maßnahme angesehen werden.

Die Analyse an sich besteht aus dem Sammeln von Informationen über eine binäre Datei, ohne dass sie dabei in irgendeiner Weise ausgeführt wird. Unabhängig von den Mitteln, die man zur Lokalisierung und Identifikation eventuell verdächtiger oder schädlicher ausführbarer Dateien einsetzt, beinhaltet der erste Schritt eine ausführliche Dokumentation, nämlich: auf welchem System befand sich die Datei, was war ihr kompletter Pfad und wer (und wann) sie gefunden und wie gesichert hat. Je vollständiger die Dokumentation ist, desto besser.

Eines der ersten Schritte bei der Analyse von Binärdateien ist es, wie im Abschnitt 6.3.1.4 bereits beschrieben, die verdächtige Datei mit Antivirus-Software zu scannen. Dieser Schritt bildet einen hervorragenden Ausgangspunkt. Der Antivirus-Scan vermag allerdings eventuell nichts Definitives zu erbringen.

Der nächste, optionale Schritt besteht aus einer String-Analyse der Datei - dabei werden, ähnlich wie bei der Hauptspeicheranalyse (6.3.1.1.5], alle ASCII- sowie UNICODE-Strings aus der Datei extrahiert - unter Umständen lassen sich daraus wichtige Informationen, wie z.B. Funktionalität des Tools oder Informationen über Netzverbindungen ableiten.

Anschließend können die vom Programm geladenen DLL-Bibliotheken analysiert und Informationen, die eventuell für eine Untersuchung nützlich sein könnten, extrahiert werden. Die meisten Programme setzen auf die Windows-API auf und nutzen verschiedene Funktionen, die von den System-DLLs zur Verfügung gestellt werden. Deshalb sollte im Rahmen einer forensischen Untersuchung überprüft werden, welche DLLs und Funktionen von diesem Programm angesprochen werden. Diese Informationen werden in der IMPORT-Tabelle und der Importadressentabelle (IAT) der ausführbaren Datei bereitgestellt.

Das Programm Dependency Walker [Mil] bietet einen einfachen Zugang zu den Informationen in der IMPORT-Tabelle - so kann festgestellt werden, welche Funktionen eine Anwendung importiert und was sie während der Ausführung macht. Zum Beispiel, wenn eine Applikation DLLs, die Netzwerk-Code enthalten, importiert (so wie in der Abbildung 6.17 zu sehen), ist es wahrscheinlich, dass die Anwendung eine Hintertür einrichtet oder dazu verwendet werden kann, um weitere Daten aus dem Internet nachzuladen.

Die Analyse der IMPORT-Tabelle kann also ansatzweise Klarheit schaffen, ob es sich bei der entsprechenden Datei tatsächlich um Malware oder um ein harmloses Programm handelt. Zum Beispiel, wenn die IMPORT-Tabelle ausschließlich *KERNEL32.DLL* sowie zwei oder drei weitere DLLs referenziert und nur wenige importierte Funktionen aus *KERNEL32.DLL*, beispielsweise *LoadLibraryA* und *GetProcAddress*, beinhalten, so weist das darauf hin, dass die Datei in gewisser Weise verschleiert wurde [Car09].

Malware-Autoren sind oft stark bemüht, ihre Dateien vor Entdeckung durch Antiviren-Software sowie IT-Administratoren zu schützen. Sie nutzen hierfür unterschiedliche Techniken, nämlich *Cryptor, Binder* und *Packer*.

„Cryptor“ ist eine umgangssprachliche Bezeichnung für Programme, die es den Malware-Autoren ermöglichen, ihre Programme zu verschlüsseln. Das Verschlüsseln einer ausführbaren Datei ist eine derzeit sehr beliebte Methode, um der Entdeckung durch sowohl Host- und als auch Netz-basierten Schutzsysteme zu entgehen.

„Binder“ sind Hilfsprogramme, die es dem Angreifer ermöglichen, eine schädliche Anwendung in einer anderen, scheinbar harmlosen Anwendung zu verstecken.

„Packer“ ist eine Bezeichnung für Programme, die es dem Angreifer ermöglichen seine Programme komprimieren zu lassen, um Platz zu sparen. Packers machen die Analyse von ausführbaren Dateien schwieriger.

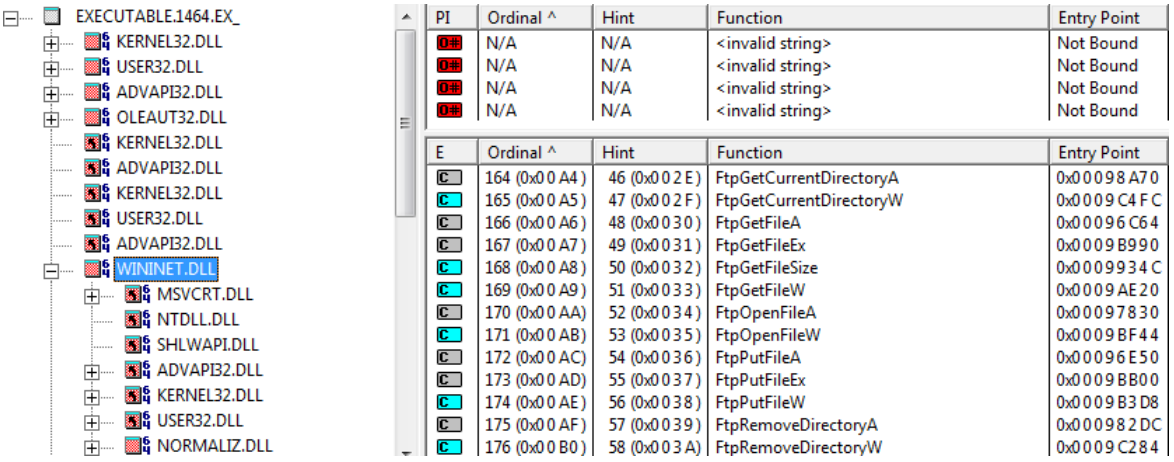

Abbildung 6.17.: Dependancy Walker listet alle referenzierten Bibliotheken und Funktionen übersichtlich auf.

Malware-Autoren nutzen sehr häufig mindestens eine der oben genannten Vorgehensweisen, um ihre Software vor Entdeckung zu schützen, oder sie passen ihre Software geringfügig an - sie ändern beispielsweise die Byte-Anzahl oder das Segment-Offset, sodass die Software keiner bekannten Signatur mehr ähnelt und der Schutz nicht mehr greift.

Weiterhin kann die Auswertung der PE-Headers zusätzliche Informationen über die ausführbare Datei liefern, wie z.B. das Datum, an dem die Anwendung kompiliert wurde. Da diese Informationen leicht vom Angreifer manipuliert werden können, wird an dieser Stelle auf eine ausführliche Behandlung verzichtet. Bei Microsoft ist eine detaillierte Dokumentation des PE-Headers erhältlich [Zbi].

6.3.2. Forensische Analyse unter SLES 11

Dieser Abschnitt beschreibt die wesentlichen Methoden einer systematischen Port-Mortem-Analyse für SUSE Linux Enterprise Server. Anschließend wird basierend auf den beschriebenen Analyse-Maßnahmen eine Sammlung von etwa 20 durchzuführenden ersten Schritten eingeführt.

6.3.2.1. Arbeitsspeicheranalyse

Gründe für die Initiierung: Suche im Hauptspeicher nach Malware und Spuren, Programmcode sowie Programmmodule extrahieren, Angriffsquelle bestimmen.

Verantwortlich für Umsetzung: IT-Forensiker

Geschätzter Aufwand: mittel bis hoch

Im Laufe des letzten Jahres wurden gängige IT-Forensikprozesse um die Arbeitsspeicheranalyse unter Linux erweitert. Durch diese zusätzliche Auswertung können viele Vorgänge, wie z.B. Rootkit-Erkennung, vereinfacht werden, und die forensische Analyse kann zielführender erfolgen. Eine der größten Herausforderungen hierbei ist die Verfügbarkeit der entsprechenden Tools, die im LRZ-Umfeld eingesetzt werden können. Während für Windows eine Reihe von Analysewerkzeugen existieren [6.3.1.1], ist die Auswahl für Linux, bedingt durch die Systemarchitektur, sehr eingeschränkt - eine Speicheranalyse besteht aus der sequentiellen Durchsuchung des Hauptspeicherabbilds nach bekannten Kernel-Strukturen. Werden sie identifiziert, können Informationen aus dem RAM-Abbild extrahiert werden. Weiterhin variiert, je nach Kernel-Version, auch der Aufbau der geladenen Objekte - Informationen können nicht (oder nicht korrekt) ausgelesen werden, falls die Mustererkennung zum Auffinden von Datenstrukturen im Speicher versagt [Ric].

Einzig das in Python geschriebenes Forensik-Tool „Volatility Framework“ verspricht derzeit eine breite Unterstützung von Kernel-Versionen - bei Bedarf können auch eigene Profile durch das Kompilieren eines Debug-Kernels und die Extraktion der Kernel-Strukturen mit dem Tool *dwarfdump* [A.] erstellt werden - Flexibilität, aktive Weiterentwicklung und Unterstützung von einer breiten Palette von Kernel-Variationen sind wichtige Faktoren für das Leibniz-Rechenzentrum. Der Erstellungsprozess besteht typischerweise aus der Kompilierung und dem Linken des Debug-Kernels, der Source-Code liegt typischerweise im *usr/src/linux* Verzeichnis. Bei diesem Vorgang wird die Datei vmlinux angelegt. Im zweiten Schritt werden mit dem Befehl *dwarfdump -di vmlinux-file >suse_2_6_32.out* DWARF-Debug-Information aus den ELF-Objekten extrahiert und in einer neuen Datei angelegt. Um eine Signaturdatei für Volatility zu generieren, muss ein Pythonskript aufgerufen werden: *python tools/dwarfparse.py -s /usr/src/linux suse_2_6_32.out >suse_2_6_32_vtypes.py* und die neuangelegte *vtype* Datei in das Verzeichnis *volatility\plug-ins\overlays\linux* kopiert werden. Schließlich muss noch eine entsprechende Profildatei im Ordner *volatility\plugins\overlays\linux* erstellt werden. Weiterführende Informationen zu diesem Thema sind bei [Ric] zu finden.

Ein weiteres Problem besteht darin, dass die jetzige Version dieses Forensik-Toolsets sich momentan noch im Alphastadium befindet - immerhin werden bereits auch 64-bittige Linux-Varianten unterstützt. Es darf davon ausgegangen werden, dass im Laufe von 2012 eine ausgereifte(re) Ausführung veröffentlicht wird [Aut]. Deswegen soll an dieser Stelle nur allgemein auf die Möglichkeiten der Linux-Hauptspeicher-Analyse eingegangen werden. Bei den zukünftigen Revisionen des Leitfadens, sollte das große Potential der Hauptspeicheranalyse größere Berücksichtigung finden: damit lassen sich Spuren aufdecken, ohne dass ein Rootkit sich davor verstecken könnte.

Die Vorgehensweise bei der Analyse von Hauptspeicherabbildern lässt sich in mehrere Bereiche unterteilen - sie entsprechen in etwa der Aufteilung bei der Live-Response [5.6.2], es werden gleiche relevante Fragestellungen beantwortet. Grundlagen zur Erstellung eines Speicherabbilds wurden bereits im Abschnitt 5.6.2.9 vermittelt.

In der Zukunft, sobald eine verlässliche Möglichkeit zur Arbeitsspeichersicherung sowie eine stabile Analyseumgebung zur Verfügung steht, könnte Arbeitsspeicheranalyse die Live-Response fast oder komplett ablösen.

Es können folgende Teilbereiche der RAM-Forensik unter Linux unterschieden werden:

- Untersuchung der laufenden Prozesse,

- Analyse der Netzverbindungen,
- Analyse des Kernel-Informationen,
- Erkennung von Rootkits.

Im Wesentlichen ist die derzeitige Funktionalität überschaubar - die Informationen, die extrahiert werden können, lassen sich auch während des Live-Response sammeln. Der Vorteil der RAM-Forensik liegt auf der Hand: Schutzmechanismen der Malware werden, unabhängig von deren Komplexität, ausgehebelt. Folgende Funktionen stehen derzeit zur Verfügung:

6.3.2.1.1. Prozessanalyse

Unter Linux sind die Speicherbereiche verschiedener Prozesse über doppelt verkettete Listen *task_struct* miteinander verknüpft. Bei der RAM-Analyse werden die beiden Zeiger *next* und *prev* ausgewertet und alle aktiven, gelisteten Prozesse erkannt. Der entsprechende Befehl lautet **linux_task_list_psaux** - neben der Auflistung aller Prozesse werden auch die Aufrufparameter aus dem Speicherimage extrahiert.

Weitere Anhaltspunkte über eventuell aktive Hintertürprogramme finden sich in der Auflistung des aktuellen Prozess-zu-Ports-Mappings. Das Plugin **linux_netstat** zeigt die komplette Socketliste der Prozesse an, die zum Zeitpunkt der Abbilderstellung auf dem IT-System gestartet waren. So lassen sich Schadprogramme aufspüren, die Daten übers Netz übermitteln.

Weiterhin lassen sich zusätzliche Informationen durch die Auswertung von *kmem_cache* extrahieren. *kmem_cache* ist ein Bestandteil des Slab-Allokators, einem Verwaltungsmechanismus für das Alloziieren von Hauptspeicher[4] und dient als Cache für Objekte, die häufig benutzt werden. Die Bereitstellung der Speicherbereiche mit diesem Algorithmus hinterlässt Artefakte, da der Allokator dazu tendiert, zurückgegebene Speicherbereiche nicht sofort freizugeben - Informationen über bereits beendete Prozesse lassen sich unter Umständen aus dem *kmem_cache* auslesen - dies hängt davon ab, welcher Allokator-Typ verwendet wird und wie schnell reservierte Speicherbereiche wieder freigegeben werden [Mau03]. Der entsprechende Befehl lautet **linux_tasklist_kmem_cache**.

6.3.2.1.2. Netzverbindungen

Für die Analyse von Netzverbindungen stehen eine Reihe von sinnvollen Maßnahmen zu zur Verfügung. Sie entsprechen bis auf eine Ausnahme in ihrer Funktionsweise den im Abschnitt 5.6.2.4 vorgestellten Verfahren. Die Plugins, die genutzt werden können, sind:

- **linux_arp**: Extraktion der aktuellen ARP-Tabellen,
- **linux_ifconfig**: Anzeige der gegenwärtigen Netzwerkadapter-Konfiguration,
- **linux_route**: Anzeige der Routingtabellen,
- **linux_route_cache**: Extraktion der Informationen aus der Forwardingtabelle (auch bekannt als Forwarding Information Base) - sie enthält zuletzt genutzte Routingeinträge. Wenn ein Paket weitergeleitet soll, werden zunächst Einträge aus dieser Tabelle überprüft, bevor die Routingtabellen aufgerufen werden. Daraus lassen sich Angaben über die letzten Verbindungsversuche ableiten. Unter Umständen lassen sich auch Anzeichen für Manipulationen erkennen [MICa].

6.3.2.1.3. Kernel-Informationen

Während der RAM-Analyse sollten, sofern nicht bereits während der Live-Response geschehen [5.6.2.2], bestimmte Kernel-Informationen aus dem Hauptspeicher extrahiert werden, nämlich:

[4] Dabei wird der vorhandene Speicherplatz in kleine Einheiten unterteilt, um Speicherfragmentierung sowie Speicherplatzverschwendung zu unterbinden.

- **linux_dmesg**: Inhalt des Ringpuffers - das Laden und Entladen der Kernel-Module lassen sich damit ausfindig machen,
- **linux_lsmod**: Auflistung aller geladenen Kernel-Module. Ein Angreifer kann Rootkit-Module in den Kernel laden - sie lassen sich auf diese Weise identifizieren.

6.3.2.1.4. Erkennung von Rootkits

Die Analyse des Arbeitsspeichers bietet völlig neue Perspektiven bei der Erkennung von Rootkits. Dieser Ansatz ist vor allem dann sinnvoll, wenn angenommen werden kann, dass das IT-System von einem LKM-Rootkit befallen wurde. Typischerweise unterscheidet man bei der Speicheranalyse zwei Unterarten der Rootkits:

- statische Rootkits,
- dynamische Rootkits.

statische Rootkits:

Vereinfacht dargestellt, tauschen solche Rootkits Systemfunktionen, globale Datenstrukturen bzw. IDT (Interrupt Descriptor Table) direkt im Speicher aus. Diese sind dann so modifiziert, dass Systembefehle keine Rootkit-Objekte mehr anzeigen. Malware könnte den Systemaufruf stat modifizieren, um Rootkit-Dateien vor den Anti-Rootkit-Programmen zu verstecken, die im Benutzer-Modus laufen. Ähnliches gilt für das Adore-Rootkit, das readdir-Funktionalität für die proc- und root-Dateisysteme modifiziert, in deren Ausgabe Rootkit-Dateien nicht mehr auftauschen. Es existieren noch weitere, bekannte Ansatzpunkte, die von Malware-Autoren gerne benutzt werden - so könnten Modifikationen an Systemfunktionen vorgenommen werden, die aus dem proc-Dateisystem Informationen über geladene Module, Netzverbindungen, geöffnete Dateien und ähnl. extrahieren. Die Erkennung solcher Schädlinge basiert auf dem Vergleich der Debuginformationen aus dem vorkompilierten Kernel (vmlinux) mit den Datenstrukturen im Hauptspeicher - Abweichungen sind ein Indiz für das Vorhandensein eines Rootkits. Weiterhin lassen sich modifizierte Strukturen genau identifizieren und somit die Unterart des schadhaften Codes bestimmen. Diese Stellen werden bei der Kernel-Profil-Erstellung für das Volatility Framework für die jeweilige Kernel-Version identifiziert und während des Analysevorgangs auf Anzeichen von Malware überprüft.

dynamische Rootkits:

Es handelt sich hierbei um eine modernere Gattung der LKM-Rootkits. Hier werden nicht statische, sondern dynamische Strukturen verändert - Informationen werden aus Listen, Hash-Tabellen sowie weiteren Datenstrukturen entfernt. Die Analyse mittels der Syscall-Tabelle oder vmlinux bleibt ergebnislos, erst eine Überprüfung der kmem_cache-Liste kann Hinweise auf schadhaften Code liefern - existieren Datenstrukturen lediglich in kmem_cache, so handelt es sich vermutlich um einen Schädling [Cas].

6.3.2.2. Analyse der Systemlogs

Gründe für die Initiierung: Art des Angriffs ermitteln, Angriffsquelle & Schwachstelle bestimmen, Ausmaß des Schadens quantifizieren, Angriffszeitpunkt ermitteln, Korrelation der Daten.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: hoch

Linux Betriebssysteme haben eine Vielzahl von Logdateien, die wichtige Hinweise während einer Untersuchung geben können. Nicht nur die Systemaktivitäten wie Benutzeran- sowie Abmeldungen werden protokolliert, sondern auch Ereignisse, die mit SUSE Linux Netzwerk-Services verknüpft sind. Analyse von Logdateien unter SUSE Linux ist in der Regel recht unkompliziert, da die meisten Protokolle im Klartext gespeichert werden - mit einer Zeile pro Ereignis. Die Verarbeitung der Protokolldateien wird in der Regel über Kommandozeilen-Tools durchgeführt.

6.3.2.2.1. Primäre Protokolldateien

Der Großteil der System-Logs auf einem Linux-System sind im „/var/log“-Verzeichnis gespeichert - entweder im Hauptverzeichnis oder in den von einzelnen Anwendungen generierten Unterverzeichnissen (z.B. „/var/log/apache2“). Da Linux System-Logs in der Regel im reinen Textformat gespeichert sind, bietet es sich an Linux-Tools für die Analyse zu verwenden - eine Verkettung einer Handvoll von Textverarbeitungs- und Suchfunktionstools macht eine einfache Datenreduktion auf wesentliche Ereignisse möglich. Die wichtigsten Tools für die Logdatei-Verarbeitung sind *sed, awk* und *grep* - sie sind auch unter Windows innerhalb der Cygwin- bzw. GnuWin32-Umgebung lauffähig.

Zur Überprüfung von Logdateien gehören mehrere Elemente. Typischerweise sollte man nach folgenden Auffälligkeiten Ausschau halten [CER]:

1. Unerwartete Neustarts von Diensten - speziell von syslog,
2. Ungewöhnliche Fehlermeldungen oder Abstürze von Diensten,
3. Steuerzeichen oder sehr lange Einträge (so wie im Listing 6.7 zu sehen) in den Logfiles können auf einen Versuch zum Auslösen eines Buffer Overflows hinweisen.

```
Apr 8   07:47:26    localhost   SERVER[5151]:  Dispatch_input: bad request line
BBàóÿ¿áóÿ¿âóÿ¿ãóÿ¿XXXXXXXXXXXXXXXXXX0000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000480000000173835088security0000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000RlFqTy/AbEnjlCM@^@^@^@^@@URlFqTy/AbEnjdb6lCMtvfx7hM2Bbs4tU2B
CCi$RvrBkBEUKXp2EujObLHck0uvmdzbz8R9^@^@^!^hLUx5JJIOhJ9jB1GvCAx3fK6lbNlC/bin/sh
```

Listing 6.7: Auszug aus der „/var/log/messages“ Datei [CM08]

Abhängig von der Art des Sicherheitsvorfalls muss eine Reihe von Logdateien inspiziert werden. Tabelle 6.4 listet alle Protokolldateien und ihr Speicherort.

Logdatei	Beschreibung
%SystemRoot%/var/log/messages	primäre Logdatei
*%SystemRoot%/var/log/apache2/**	Apache-Logs
*%SystemRoot%/var/log/mysql/**	MySQL-Logs
*%SystemRoot%/var/log/tomcat6/**	Tomcat-Logs

Tabelle 6.4.: Speicherorte der Protokolldateien auf der Festplatte

Abhängig von der Systemkonfiguration können Protokolldateien auch in einem anderen Verzeichnis bzw. auf einem anderen Datenträger abgelegt werden - dies gewährleistet einen bedingten Schutz gegenüber einfacheren Löschskripten. Bei der Einrichtung von kritischeren System wird am Leibniz-Rechenzentrum auf den Mechanismus des Remote-Loggings zurückgegriffen. Die Wahrscheinlichkeit der Manipulation durch den Angreifer ist weitaus geringer.

Gescheiterte und erfolgreiche Anmeldeversuche werden in der Standardkonfiguration ebenfalls aufgezeichnet. Zusammen mit den Anmeldeversuchen für Netzwerk-Dienste wie FTP oder ssh, sind Konsolen-Anmeldungen auch in der Datei „/var/log/messages“ gespeichert.

6.3.2.2.2. Benutzeraktivitätsprotokoll

Direkte Aufzeichnungen der Benutzeraktivitäten auf einem Linux-System werden in drei primäre Dateien hinterlegt, nämlich „/var/run/utmp“, „/var/log/wtmp“ sowie „/var/log/lastlog“ - unter Umständen wurde ihr Inhalt bereits während des Live-Response (vgl. Abschnitt 5.6.2.3) sichergestellt. Anstelle des bei Systemprotokolldateien üblichen reinen Textformats nehmen die „utmp“ und „wtmp“-Dateien die An-und Abmeldungeinformationen der Benutzer in einem binären Format auf. Der wesentliche Unterschied zwischen diesen Dateien

ist, dass „utmp“ nur Informationen über aktive Systemanmeldungen hält, während „wtmp“ langfristig Logon-Informationen speichert. „lastlog“ ist eine Binär-Lodatei, die die letzte Anmeldezeit und den Remote-Host für jeden Benutzer auf dem System speichert. Es sollte beachtet werden, dass nicht alle An- und Abmeldungen, abhängig vom verwendeten Programm, protokolliert werden und Hintertüren in der Regel keine Einträge in den besagten Logdateien generieren [CM08].

6.3.2.2.3. Shell-Historien

Benutzer mit interaktivem Zugriff auf SLEX-Systeme nutzen eine Kommandozeilenumgebung, in der Regel handelt es sich um die Bourne-Again (bash) Shell. Sie bietet die Möglichkeit, alle Befehle zusammen mit ihren Kommandozeilen-Optionen zu protokollieren - so kann bei Bedarf zurückverfolgen werden, welche Schritte ein Benutzer während der letzten Sitzung in der Shell durchgeführt hat. Typischerweise wird die History-Datei als versteckte Datei „.bash_history“ in dem Basisverzeichnis des Benutzers gespeichert - sie kann wertvolle Informationen liefern, sofern sie während des Eingriffs nicht manipuliert oder gelöscht wurde. Ein weiteres Problem betrifft das Fehlen von Zeitstempeln - „.bash_history“ kann zwar eine Auflistung der Shell-Kommandos liefern, es lassen sich keine Aussagen über den zeitlichen Ablauf machen [CP08].

6.3.2.2.4. SSH-Protokolle

Gründe für die Initiierung: SSH-Angriffe.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren, Netz-Administratoren

Die Dateien, die im direkten Zusammenhang mit dem SSH-Dienst stehen und im Ordner *.ssh* abgelegt werden, können oft sehr aufschlussreich sein. Bei jedem Verbindungsversuch von einem entfernten System, wird ein Eintrag in der Datei *.ssh/known_hosts* erstellt. Die generierten Informationen lassen sich wie folgt gliedern:

1. IP-Adresse des entfernten Systems,
2. SSH-Public-Key.

Diese Informationen können mit anderen Quellen korreliert werden zur Feststellung, ob noch weitere Systeme in der Umgebung betroffen sind.

6.3.2.3. Aufspüren von nichtautorisierten Nutzer- oder Gruppenkonten

Gründe für die Initiierung: Konfigurationdateien auf das Vorhandensein neuer Benutzer überprüfen.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: gering

Die gezielte Modifikation der Account- und Gruppeninformationen zählt zu den grundlegenden Maßnahmen bei einem Cyber-Angriff, sei es durch das Hinzufügen von zusätzlichen Benutzerkonten oder aber durch das Anpassen der Rechte bereits bestehender User. In der Praxis soll dadurch in erster Linie eine Hintertür für zukünftige Zugriffe auf das Ziel-System eingerichtet werden. Angreifer sind - leider - in der Regel erfolgreich, somit ist es für praktische Belange wichtig die zugrunde liegende Struktur der entsprechenden Konfigurationsdateien zu verstehen, um bei einer Überprüfung feststellen zu können, ob sich ein Angreifer daran zu schaffen gemacht hat. Wichtig ist dabei die Erkenntnis, dass dies ein recht unkomplizierter Vorgang ist, sofern der Angreifer seine Spuren nicht verwischt hat.

Die Suche nach möglichen Hinweisen bei den Benutzerkonten beginnt in der Datei „/etc/passwd“, die eine Liste aller Nutzer sowie die vollständigen Pfade derer Heimverzeichnisse beinhaltet. Ein typischer Eintrag besteht aus sieben Feldern mit möglichen Sonderzeichen:

1. Benutzername,
2. Kennwort (wird nicht mehr verwendet),

3. User-ID,
4. Group-ID,
5. optionale Beschreibung,
6. vollständiger Pfad zum home-Verzeichnis,
7. vollständiger Pfad zur ausführbaren Datei, die nach einem erfolgreichen Login ausgeführt wird.

Ein Beispiel kann dem nachfolgenden Listing 6.8 entnommen werden.

```
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
dnsmasq:x:104:65534:dnsmasq:/var/lib/empty:/bin/false
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
ftpsecure:x:109:65534:Secure FTP User:/var/lib/empty:/bin/false
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
mysql:x:60:110:MySQL database admin:/var/lib/mysql:/bin/false
root:x:0:0:root:/root:/bin/bash
sshd:x:101:102:SSH daemon:/var/lib/sshd:/bin/false
tftp:x:108:106:TFTP account:/srv/tftpboot:/bin/false
wwwrun:x:30:8:WWW daemon apache:/var/lib/wwwrun:/bin/false
lrz:x:1000:100:lrz:/home/lrz:/bin/bash
(...)
```

Listing 6.8: Inhalt der „/etc/passwd“ Datei

Die Überprüfung kann auf unterschiedlichen Ebenen ansetzen. Ein Nutzerkonto mit der *UID 0* besitzt praktisch dieselben Rechte wie die Rootkennung - sollte irgendein Benutzerkonto ausgeweitete Zugriffsrechte aufweisen, verbirgt sich dahinter mit hoher Wahrscheinlichkeit eine Hintertür. Im Rahmen der Analyse sollte „/etc/passwd“ grundsätzlich auch auf das Vorhandensein neuer Nutzer überprüft werden [CP08].

Durch die systematische Auswertung der Gruppeninformationen können Grundlagen für spätere Schritte gewonnen werden - daher sollten die Einträge in „/etc/group“ überprüft werden. Diese Datei besitzt ein der „/etc/passwd“ sehr ähnliches Format, wenn auch mit weniger Feldern. Für gewöhnlich sieht der Eintrag der „/etc/group“ folgendermaßen aus:

1. Groupenname,
2. Kennwort (wird in der Regel nicht verwendet),
3. Group-ID,
4. Group-List - eine Auflistung aller der Gruppe angehörigen Mitlieder.

Die Datei „/etc/group“ listet sämtliche Gruppen, deren IDs sowie alle der jeweiligen Gruppe zugeordneten Nutzer auf. Sollten hier zusätzliche nicht-autorisierte Nutzer innerhalb privilegierter Gruppen (z.B. *root* oder *wheel*) auftauchen, sind diese verdächtig und erfordern weitere Nachforschungen (vgl. Listing 6.9) [CP08].

```
at:!:25:
bin:x:1:daemon
daemon:x:2:
dialout:x:16:
ftp:x:49:
kmem:x:9:
man:x:62:
mysql:!:110:
nogroup:x:65534:nobody
public:x:32:
root:x:0:
shadow:x:15:
sshd:!:102:
tftp:!:106:tftp
utmp:x:22:
wheel:x:10:
www:x:8:
users:x:100:
(...)
```

Listing 6.9: Inhalt der „/etc/group“ Datei

„etc/shadow“ ist die dritte Datei, die für die Authentifizierung innerhalb von SLEX-Systeme zuständig ist. Diese enthält neben gehashten Benutzerpasswörtern weitere Passwort-relevante Informationen:

1. Benutzername,
2. Kennwort,
3. Datum der letzten Kennwortänderung,
4. Minimale Gültigkeit des Passworts, bevor eine (erneute) Kennwortänderung erlaubt ist,
5. Maximale Gültigkeit des Passworts,
6. Anzahl der Tage vor Ablauf der Passwortgültigkeit, an der eine Warnung angezeigt wird,
7. Tagesanzahl nach Ablauf der Lebensdauer, nach der ein Account deaktiviert wird,
8. Tag der Deaktivierung des Accounts (gezählt ab 1.1.1970).

Das folgende Listing zeigt einen Beispielausschnitt aus „/etc/shadow“:

```
at:*:15322:0:99999:7:::
bin:*:15288::::::
daemon:*:15288::::::
ftp:*:15288::::::
ftpsecure:*:15322:0:99999:7:::
mysql:*:15322:0:99999:7:::
root:$6$oN1MaU/T$nJFnG8xD1oq.RlFqTy/AbEnjlCMtvfx7hM2Bbs4tU..
    bpJjj8831a3aNTXJ26kPvJTwP/aIqQkyuDnYWSgxNo1:15322::::::
sshd:*:15288:0:99999:7:::
tftp:*:15322:0:99999:7:::
wwwrun:*:15288::::::
lrz:$6$1neb.
    CCi$RvrBkBE1UKXp2EuSujObLHck0uvmdzbnz8R9Yjuj3xKXghLUx5JJIOhJ9jB1GvCAx30/
    fK6lbNlC0qCepUVah0:15322:0:99999:7:::
(...)
```

Listing 6.10: Inhalt der „/etc/shadow“ Datei

Daemon-Dienste nutzen keine Passwörter - daher werden auch keine Kennwörter vergeben, um die Nutzung dieser Kennungen für ein interaktives Login zu verhindern. Jegliche Konten, die keinem Nutzer zuzuordnen sind und trotzdem über verschlüsselte Passwortfelder verfügen, sollten unbedingt genauer untersucht werden [CP08].

6.3.2.4. Überprüfung der Jobsteuerung

Gründe für die Initiierung: Spuren für das Einnisten der Malware finden.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: gering

Um auf Linux-Systemen eine Aktion zur späteren Ausführung festzulegen, stehen zwei grundlegende Möglichkeiten zur Auswahl: „At-“ und „cron-Jobs“. Ist ein Einbruch erfolgreich, so bieten cron- und At-Jobs einem Angreifer eine einfache Möglichkeit auf dem SLEX-System bestimmte, wiederkehrende Abläufe zu automatisieren und ein IT-System langfristig ihrem Einfluss auszusetzen - sie können Logfiles bereinigen, Systemkonfiguration anpassen, die Liste ließe sich noch lange fortsetzen. Diese Aktivitäten aufzuspüren, ist für eine Ermittlung von großer Wichtigkeit.

Mit dem Befehl „at“ lassen sich Kommandos zur einmaligen Ausführung zu einem späteren Zeitpunkt festlegen. „At-Jobs“ findet man unter */var/spool/atjobs*, laufende Aufträge werden unter */var/spool/atspool* abgelegt. Mit „cron-Prozessen“ werden hingegen regelmäßig ablaufende Aufgaben festgelegt, z.B. täglich, wochentlich, oder monatlich. Diese werden an zwei unterschiedlichen Orten abgelegt: System cron-Jobs findet man in gleich mehreren Verzeichnissen, die in der Datei *etc/crontab* definiert und meistens auch entsprechend benannt sind, also z.B. */etc/cron.hourly*, */etc/cron.daily*, */etc/cron.weekly* oder */etc/cron.monthly*. Von Usern festgelegte Aufgaben befinden sich im Verzeichnis */var/spool/cron*.

6.3.2.5. „Versteckte“ Dateien

Gründe für die Initiierung: Vom Angreifer maskierte Dateien lokalisieren.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: mittel

Auf Linux-Systemen sind Dateien vor dem Blick des Betrachters „verborgen“, falls ihr Name mit einem Punkt beginnt. Diese Dateien werden unter den meisten grafischen Oberflächen sowie Kommandozeilen-Anwendungen normalerweise nicht angezeigt. Eine Möglichkeit, solche Dateien in der Kommandozeilenumgebung erscheinen zu lassen, ist der Befehl *ls -a*.

Des Weiteren versehen Angreifer Dateien und Verzeichnisse häufig mit scheinbar unbedenklichen Bezeichnungen, sei es *rpc.auditd* für einen Sniffer oder *.X11-R4* bzw. ... als Ordnername. Diese Bezeichnungen sind denen bestehender Dateien und Ordner sehr ähnlich, so dass ihre Anwesenheit in einer Verzeichnisliste oder Prozesstabelle beim Administrator unter Umständen keinen Verdacht erregt.

```
tolwyn@linux-0idh:/var/tmp> ls
kdecache-root  kdecache-tolwyn  var
tolwyn@linux-0idh:/var/tmp> ls -a
.  ..  ...  kdecache-root  kdecache-tolwyn  var
```

Abbildung 6.18.: Der Befehl *ls -a* zeigt versteckte Dateien und Verzeichnisse an.

Sinnvollerweise sollte auch das */tmp*-Verzeichnis überprüft werden. Da sämtliche Nutzer und Prozesse für diesen Ordner Schreibrechte besitzen, ist er häufig die erste Anlaufstelle für Angreifer, die das System mit schädlichem Code infizieren. Weiterhin nutzen einige Exploits das temporäre Verzeichnis für Privilegien-Eskalationsangriffe. Es ist denkbar, dass gewisse Artefakte, Verzeichnisse bzw. Dateireste vom Angreifer zurückgelassen wurden - diese könnten Hinweise auf die Aktivitäten des Hackers liefern [CP03].

Eine weitere Stelle, die häufig von Eindringlingen genutzt wird, um Dateien zu verbergen, ist das */dev*-Verzeichnis - schon deswegen, weil dieses Verzeichnis sehr viele Dateiobjekte enthält, deren Zugriffszeiten sich laufend ändern. Um Dateien aufzuspüren, die bei einem SI zurückgelassen wurden, empfiehlt es sich, die Berechtigungen aller Dateien auszuwerten - Gerätedateien werden entweder mit einem „b“ für Block-Device (blockorientiertes Gerät) oder einem „c“ für Character-Device (zeichenorientiertes Gerät) gekennzeichnet.

6.3.2.6. Überprüfung der Konfigurationsdateien

Gründe für die Initiierung: Einschätzung des Schadens, Ausmaß der lokalen Modifikationen bestimmen.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: mittel

Eine weitere Fundstelle von Beweisen bei forensischen Untersuchungen stellen Konfigurationsdateien dar. Jeder Dienst, der auf dem System aktiv ist, besitzt für gewöhnlich mindestens eine solche Datei (manchmal jedoch auch mehrere), die sein Verhalten bei der Ausführung steuern. Für einen versierten Eindringling ist es eine Leichtigkeit, die Konfigurationen von Anwendungen zu modifizieren und sich zunutze zu machen. Häufige Anlaufstellen sind die Dateien, die für den Zugang zum IT-System zuständig sind. Angreifer können diese Dateien so modifizieren, dass anderen Computern der Zugang zum Zielsystem jederzeit ermöglicht wird.

Die xinetd-Daemon Konfigurationsdatei *xinetd.conf*, die sich im Ordner */etc* befindet, verwaltet auf Linux-Systemen eine Vielzahl von Netzdiensten. Dienste wie z.B. *telnet, FTP* oder *TFTP* werden über jene Datei gestartet. Durch zusätzliche Einträge in diese Datei ist es Angreifern möglich, ein Zielsystem durch geöffnete Ports für illegale Zugriffe angreifbar zu machen. Ebenso können zuvor inaktive Dienste wie *FTP* durch Angreifer reaktiviert werden.

6.3.2.7. Erkennung von Rootkits

Gründe für die Initiierung: Verdacht auf Malware

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: gering

Wie in jedem großen IT-Betrieb hat das LRZ-Personal häufig mit Kompromittierungen auf Basis von Rootkits zu tun. Die Folgen können sehr weitreichend sein: Rootkits bringen meistens nicht nur Hintertüren, sondern auch Tastatur-Logger, Netzwerk-Sniffer sowie andere Tools, die großen Schaden anrichten können. Aus diesem Grund ist es sehr wichtig, rechtzeitig nach Auffäligkeiten zu suchen und schadhaften Code zu identifizieren. Bei einer genauen Betrachtung von Linux-Rootkits wird deutlich, dass sie sich grob in zwei Bereiche, entsprechend ihrer Privileg-Stufen, einteilen lassen: Benutzer-Modus- sowie Kernel-Modus-Rootkits.

Die Vorgehensweise für die Erkennung von Rootkits entspricht in etwa den Möglichkeiten, die auch unter Windows zur Verfügung stehen (vgl. Kapitel 6.3.1.4). Die genaue Vorgehensweise ist abhängig vom konkreten Untersuchungsfall und ist mit dem SIC abzustimmen. Zu den möglichen Methoden zählen:

6.3.2.7.1. Post-Mortem-Analyse

- Erkennung von Rootkits mittels Hauptspeicher-Analyse (6.3.2.1.4).
- Falls kein Hauptspeicherabbild vorliegt, kann ein Datei-Scan zur Erkennung von Rootkits durchgeführt werden. Hierfür wird das zu untersuchende Abbild schreibgeschützt in das Analysesystem eingebunden und mit einem oder mehreren Anti-Malware-Lösungen untersucht. Bedingt durch die Verwendung des Schreibschutzes werden keine Änderungen an den Dateien vorgenommen und die Beweisintegrität bleibt erhalten. Wurde das Rootkit modifiziert oder ist es relativ neu, so besteht die Gefahr, dass es so nicht gefunden werden kann. Die gleiche Problematik greift auch dann, wenn es sich um einen Speicher-Rootkits handelt.

6.3.2.7.2. Live-Analyse

Bei einer Rootkit-Analyse eines lebenden Systems lassen sich grundsätzlich zwei mögliche Vorgehensweisen identifizieren:

- Erkennung von Rootkits mittels Datenablgeich. Die Idee dahinter ist simpel - Daten aus zwei verschiedenen Quellen werden erfasst und miteinander verglichen. Oft lassen sich zur Laufzeit eines Systems dadurch Auffälligkeiten bzw. Diskrepanzen erkennen, die auf schadhafte Software hindeuten. Ein Beispiel wäre ein lokaler und ein externer Port-Scan - an dieser Stelle sei auf den Abschnitt 5.6.2.4 verwiesen.
- Durchführung eines spezifischen Datei-Scans zur Erkennung von Rootkits. Die Analyse eines noch aktiven, nicht ausgeschalteten Systems fällt streng genommen nicht in den Bereich der Computer-Forensik - es werden viele Artefakte auf dem Host hinterlassen und die Integrität der Beweismittel kann nicht garantiert werden. Dieser Ansatz ist sinnvoll, wenn das System aus Verfügbarkeitsgründen nicht ausgeschaltet werden kann oder aber eine forensische Untersuchung nicht im Vordergrund steht. Es existieren eine Reihe von Tools, die nach nach Rootkits, Hintertüren und lokalen Exploits suchen, von denen rkhunter [Boe] und chkrootkit [Mur] die Bekanntesten sind. Abhängig von der Malware-Art kann eine solche Untersuchung ergebnislos bleiben.

6.3.2.8. Timeline Analyse

Gründe für die Initiierung: Einschätzung des Schadens, zeitlichen Verlauf des Angriffs rekonstruieren, Hinweise auf weitere Beweisfundorte finden.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren, Netz-Administratoren

Geschätzter Aufwand: hoch

Auf der Suche nach Dateien und Verzeichnissen, die im Zeitraum eines Security Incidents geöffnet, verändert oder erstellt worden sind, ist es natürlich von großem Nutzen, wenn der Zeitpunkt des Zwischenfalls bekannt ist. Sollte ein IDS den Angriff entdeckt und protokolliert haben, kann der Zeitpunkt sehr genau benannt werden. In anderen Fällen, wenn Hinweise auf einen Zwischenfall möglicherweise erst mehrere Tage später entdeckt werden, lässt sich der Tatzeitpunkt allenfalls grob abschätzen.

Ziel der Überprüfung sämtlicher MAC-Stempel ist es, festzustellen, welche Dateien während des Angriffs gelesen, geschrieben, ausgeführt oder verändert wurden. So können z.B. modifizierte bzw. neuerstellte Konfigurationsdateien entdeckt werden - dies sind wichtige Hinweise für die Bestimmung des Angriffsvektors. Vom besonderen Interesse ist dabei der Zeitstempel der Kommandozeilenumgebung - sie wird häufig bei Buffer-Overflow-Angriffen aufgerufen, was ihre MAC-Zeit ändert [Ges10].

Gelegentlich besteht das Bedürfnis, sog. Zeitlinien zu erstellen - Systemaktivitäten und Bewegungen des Angreifers lassen sich auf diese Weise nachweisen, geeignete Software (wie das bereits mehrmals erwähnte *The Sleuth Toolkit* (*tsk_gettimes*) [Carc]) vorausgesetzt. Damit lässt sich bei Bedarf auch der Zeitpunkt eines SI genauer eingrenzen, indem die Aktivitätshöhepunkte des IT-Systems bestimmt werden. Auch hier gilt die Regel, dass durch den Einsatz von verschiedenen Dateisystemen, wie z.B. EXT3, EXT4, XFS oder Reiser, die Erstellung solcher Zeitlinien nur unter gewissen Bedingungen möglich ist.

Falls das verdächtige Dateisystem von *The Sleuth Toolkit* nicht unterstüzt wird, kann ein forensisches Duplikat im schreibgeschützten Modus an ein Linux-System angeschlossen werden - MAC-Time-Informationen lassen sich dann mit dem Tool *mac-robber* [Cara] sammeln. *mac-robber* nutzt dabei das Betriebssystem-API, um auf Inhalt eines Datenträger zuzugreifen.

6.3.2.9. /proc Dateisystem

Gründe für die Initiierung: Einschätzung des Schadens, Überprüfung auf Hintertüren und andere, vom Angreifer hinterlassene, Programme.

Verantwortlich für Umsetzung: IT-Forensiker, System-Administratoren

Geschätzter Aufwand: hoch

Das /proc-Verzeichnis (auch bekannt als Prozessdateisystem) ist genau genommen ein hierarchisches virtuelles Dateisystem, welches im Hauptspeicher liegt und als eine Schnittstelle zum Kernel fungiert. Aus dieser Struktur lassen sich viele Informationen über aktuelle Systemzustände sowie laufende Prozesse gewinnen und ihr Inhalt sollte während der Live-Response (siehe Abschnitt 5.6.2.5) unbedingt gesichert werden.

Zu den wichtigen Informationsquellen zählen:

cmdline: enthält die Kommandozeilenparameter, mit denen das System gestartet wurde.

kallsyms: enthält die Symboltabelle des Kernels. Einige Rootkits hinterlassen Artefakte und lassen sich so nachweisen. Im Umkehrschluss bedeutet das, dass das Fehlen von Spuren in kallsyms keineswegs die Kompromittierung des Systems ausschließt.

modules: enthält die Auflistung aller aktuell geladenen Kernel-Module. Einfachere Rootkits lassen sich mit dieser Technik nachweisen.

Informationen über Prozesse sind in den jeweiligen Unterverzeichnissen zu finden, die als Namen die Prozess-ID des Prozesses tragen. Grundsätzlich steht eine Fülle an Informationen über ausgeführte Prozesse zur Verfügung, von denen besonders folgende nennenswert und interessant sind [CP08]:

- */proc/%PID%/cmdline*: zeigt die Parameter an, die dem Prozess zum Startzeitpunkt übergeben wurden.
- */proc/%PID%/cwd*: ein Zeiger auf das Arbeitsverzeichnis des Prozesses (auch bekannt als *current working directory*).

- */proc/%PID%/environ*: zeigt alle für den Prozess definierten Umgebungsvariablen an. In einigen Versionen des Linux-Kernels ist es möglich, durch die Verwendung von überlangen bzw. ungültigen Umgebungsvariablen einen Privilegien-Eskalation-Angriff durchzuführen und Root-Rechte zu erhalten.
- */proc/%PID%/exe*: ein symbolischer Link, das auf die zum Prozess gehörende ausführbare Datei zeigt. Auch wenn ein Angreifer nach dem Start seiner Anwendung das Binary löscht, kann man es finden und wiederherstellen.
- */proc/%PID%/fd*: eine Auflistung aller vom Prozess geöffneten Dateideskriptoren.
- */proc/%PID%/maps*: bildet den virtuellen Adressraum eines laufenden Prozesses ab.
- */proc/%PID%/root*: ein Zeiger auf das Root-Verzeichnis des Prozesses.
- */proc/%PID%/status*: zeigt aktuellen Prozessstatus sowie Informationen über den Prozeßspeicher an.

6.3.2.10. Wiederherstellung von gelöschten Dateien

Gründe für die Initiierung: Zurückgelassene Spuren auf den Datenträgern analysieren, Hinweise auf gelöschte Dateien finden.

Geschätzter Aufwand: hoch

Verantwortlich für Umsetzung: IT-Forensiker

Mitunter werden vom Angreifer Dateien oder ganze Verzeichnisse gelöscht, um Spuren zu verwischen und die Aufklärung eines Vorfalls zu erschweren. Die Wiederherstellung von Dateien stellt das LRZ-CSIRT-Team vor mehrere Herausforderungen. Grundsätzlich sind zwei Verfahren möglich:

- Rekursive Wiederherstellung von gelöschten Dateien aus einem Verzeichnis, einer Partition oder einem kompletten Datenträger. Leider existiert derzeit keine zufriedenstellende Lösung, die alle Dateisysteme abdeckt. So unterstützt *The Sleuth Toolkit* derzeit keine Datenträger, die mit EXT4, Reiser oder XFS formatiert sind. Dieser Umstand gilt auch für teure proprietäre Software. Sobald entsprechende Möglichkeiten vorhanden sind (z. B. in einer neueren Version von *The Sleuth Toolkit*), kann dieser Baustein weiter ausgebaut und in mehrere Bereiche untergliedert werden, z.B. in „Analyse von gelöschten Dateien“, „Analyse der unallozierten Datenträgerbereiche“, etc.
- Werden Programme nach dem Start gelöscht, können Binaries aus dem Verzeichnis */proc/%PID%/exe* wiederhergestellt werden. Voraussetzung ist, dass das Programm noch läuft. Nähere Informationen zu diesem Thema sind im Abschnitt 6.3.2.9 zu finden.

6.3.2.11. Analyse von unbekannten Binärdateien

Gründe für die Initiierung: Hinweise auf die Herkunft oder den Programmierer der Werkzeuge finden, Fähigkeiten des Angreifers besser einschätzen, neue Angriffsmethoden frühzeitig erkennen, Spuren finden, die zur Identifikation des Täters führen können.

Verantwortlich für Umsetzung: IT-Forensiker

Geschätzter Aufwand: hoch

Heute kursiert eine Vielzahl von unterschiedlichen Malware-Arten, die sich nicht nur autonom verbreiten und Fernsteuerungs-Zugriffe ermöglichen, sondern auch ihre Aktivitäten auf dem Wirtssystem manchmal völlig verbergen. Des Weiteren ist heutige Malware bereits in der Lage, Sicherheitsprogramme zu unterlaufen, Antivirenprogramme zu deaktivieren sowie Firewalls zu umgehen, indem sie interne Netzzugriffe auf externe Steuerungs- und Kontrollserver umleitet.

Diese umfangreichen Verschleierungsmaßnahmen der Entwickler haben einen guten Grund: Sobald die Funktionsweisen ihrer Schöpfungen einmal entschlüsselt sind, wissen Ermittler, nach welchen Hinweisen oder Mustern sie auf befallenen IT-Systemen und im Netzverkehr Ausschau halten müssen. Gerade die Fülle an

wertvollen Informationen, die aus der Analyse von Schadsoftware gewonnen wird, ist inzwischen ein integraler und unverzichtbarer Bestandteil der Untersuchung von illegalen Zugriffen und der Identifizierung von Datendiebstählen geworden. Stellenweise lassen sich auf den befallenen IT-Systemen nur wenige Beweise sichern, sodass ermittlungsrelevante Informationen direkt innerhalb der Malware zu finden sind.

Der folgende Abschnitt behandelt Vorgehensweisen, Techniken und Werkzeuge für die Durchführung einer Dateianalyse. Die Voraussetzungen unterscheiden sich nicht von der Analyse der unbekannten Dateien unter Windows (vgl. Abschnitt 6.3.1.6). Vorausgesetzt wird die grundsätzliche Fähigkeit, eine verdächtige Datei aufzuspüren und zu extrahieren. Es bietet sich an die Analyse einer Linux-Datei vorzugsweise unter einem Linux-System durchzuführen, da nur die wenigsten Windows-Tools dafür geeignet sind.

Die Menge der vorhandenen Informationen innerhalb der ausführbaren Datei hängt von der Art und Weise ab, wie sie vom Angreifer kompiliert wurde. Typische Informationsquellen sind:

- Virus-Scan,
- Dateisignatur,
- Art des Bindens,
- Schlüsselwortsuche,
- Symbol- und Debug-Informationen.

6.3.2.11.1. Virusscan

Zunächst bietet es sich an, eine Datei mit einem Antivirus-Programm zu überprüfen oder vorzugsweise, sie einer Online-Viren-Prüfung zu unterziehen. Dies kann wichtige Hinweise liefern, ob die Datei schädlichen Code enthält.

6.3.2.11.2. Dateisignatur

Eine Dateisignatur ist eine spezifische Byte-Sequenz, die zur Identifikation innerhalb des Dateiheaders verankert ist und sich unter Linux normalerweise innerhalb der ersten Bytes einer Datei befindet. Unterschiedliche Dateien weisen eine unterschiedliche Dateisignatur auf. Wird während der Incident Response oder während der Post Mortem Analyse eine verdächtige Datei entdeckt, so steht den zuständigen Ermittlern unter SLEX der Befehl *file* zur Verfügung. Neben der Identifikation der Datei liefert der Befehl weitere wichtige Informationen:

- Zielplattform,
- Art des Bindens,
- Vorhandensein der Symbolinformationen.

6.3.2.11.3. Art des Bindens

Mit einer Reihe von Tools lässt sich schnell herausfinden, ob ein Binärcode statisch oder dynamisch verknüpft ist. Das gängigste Kommando, um verwendete Bibliotheken zu bestimmen, lautet *ldd* (kurz für „list dynamic dependencies“) und indentifiziert nicht nur alle genutzen Libraries sondern auch deren zugehörige Speicheradressen.

6.3.2.11.4. Schlüsselwortsuche

Einige der wertvollsten Hinweise in einer Datei finden sich in den ihr eingebetteten Textstrings. Strings enthalten eine Vielzahl wichtiger Informationen, darunter Programmfunktionen, Dateinamen, IPs bzw. URLs, Emailadressen, Fehlermeldungen sowie vieles andere mehr. Die Untersuchung von Strings liefert bisweilen äußerst aufschlussreiche Einblicke in die Fähigkeiten einer verdächtigen Datei. Falls während der Live-Response der Prozessspeicher gesichert wurde, werden alle relevanten Informationen automatisch extrahiert.

6.3.2.11.5. Symbol- und Debug-Informationen

Symbol- und Debug-Informationen werden während eines Kompiliervorgangs erzeugt. Unter Linux werden diese Informationen im ELF-Format (Das Executable and Linkable Format) abgespeichert. Sie geben über Programmvariablen sowie Funktionsnamen Aufschluss und werden daher von Angreifern häufig entfernt.

Hat ein Angreifer eine binäre Datei nicht von Programmvariablen und Funktionsnamen „befreit“, kann ein Ermittler möglicherweise viele Einblicke in die Fähigkeiten eines verdächtigen Programms (sowie des Angreifers) erhalten. Ähnlich verhält es sich, wenn ein schädliches Programm im Debug-Modus kompiliert wird, da in diesem Fall zusätzliche Informationen in Form von Quellcode und Debug-Informationen zu finden sind.

Mit dem *nm*-Kommando werden in ausführbaren Dateien eingebettete Symbol- und Debuginformationen identifiziert und ausgelesen. Für das Sammeln zusätzlicher Symbolinformationen stellen die *nm*-Utilities eine Reihe weiterer Kommandos zur Verfügung. Um spezielle Symbole aufzuspüren, kann die Option *special-syms* verwendet werden. Ist eine binäre Datei dynamisch verknüpft, kann ein Analyst durch die Option *-D* weitere Informationen extrahieren.

Alternativ kann für die Analyse statt der nm-Utilities das frei verfügbare Programm *ObjectViewer* von Paul John Floyd [Flo] eingesetzt werden. Dank der grafischen Benutzeroberfläche kann die Aufschlüsselung der Symbolinformationen schneller erfolgen.

Angreifer setzen eine Reihe von Verfahren ein, um Dateiinhalte zu verschleiern und vor Entdeckung zu schützen: Packer, Cryptor und Wrapper (entsprechen Binder in der Windows-Umgebung), vgl. Abschnitt 6.3.1.6. Einen Schutzmechnismus unter Linux aufzuspüren, ist relativ schwierig, weil keine geeigneten Tools existieren. Falls eine Abfrage des Dateityps ungewöhnliche oder fehlerhafte Filedeskriptoren liefert, so kann dies als ein Hinweis darauf gewertet werden, dass Header und/oder gemeinsam genutzte Bibliotheken durch einen Schutzmechanismus modifiziert oder versteckt wurden [CM08].

6.4. Zusammenfassung

Mit der Anwendung der forensischen Analyse werden einzelne Aufgaben sowie komplette Abläufe zur Aufklärung von Sicherheitsvorfällen strukturiert und zum Teil automatisiert. Die Vorteile liegen auf der Hand:

- Steigerung der Effektivität,
- Praxiserprobte Verfahren,
- Exakte Abbildung der LRZ-Infrastruktur,
- Berücksichtigung neuartiger Bedrohungen.

Die selbst gesteckten Ziele wurden für Windows-Systeme erreicht. Wie häufig in der IT-Welt, stellt die heterogene Zusammenstellung sowie die Systemarchitektur der SLEX-Systeme hohe Anforderungen an die verwendeten Werkzeuge - vor allem die verschiedenen zum Einsatz kommende Dateisysteme erweisen sich als Stolperstein - selbst das weit verbreitete Ext4-Dateisystem wird derzeit kaum unterstützt. Hier muss zur gegeben Zeit eine Reevaluierung des Leitfadens und Integration zusätzlicher Werkzeuge erfolgen.

Eine weitere große Herausforderung ist das Fehlen der geeigneten Software für die Arbeitsspeicheranalyse in der SLEX-Umgebung. Die Hersteller der Forensiksoftware haben dieses Problem noch nicht erkannt, lediglich das Volatility Framework befindet sich im experimentellen Alpha-Stadium - die Anwendbarkeit konnte deswegen nicht überprüft werden. Ob dieses vielversprechende Verfahren schlussendlich eingesetzt wird, hängt von den weiteren Entwicklungen auf diesem Gebiet ab.

Da es derzeit für die LRZ-Anforderungen (vor allem für die SLEX-Umgebung) keine Pauschallösung existiert und die hohen Lizenzkosten eine Anschaffung nicht rechtfertigen, wurde auf die Erprobung der teuren Forensiksoftware verzichtet. Stattdessen wurden die bisher am LRZ genutzten Verfahren aufgegriffen, an die Anforderungen der IT-Forensik angepasst und behutsam um neue Verfahren erweitert.

Zu beachten ist, dass der Leitfaden universal und modular aufgebaut ist. Das hat den Vorteil dass sämtliche Methoden sich in fast jeder Situation anwenden lassen und dass die Incidentbearbeitung schrittweise ablaufen kann. Das verringert die Einarbeitungszeit der LRZ-Mitarbeiter erheblich.

7. Forensischer Bericht

Eine ordnungsgemäße, standardkonforme Dokumentation ist das A und O einer forensischen Ermittlung. Auch eine sorgfältige Untersuchung ist mehr oder minder wertlos, wenn die daraus resultierende Ergebnisse unzureichend dokumentiert sind. Hierfür wird eine sorgfältige Anpassung der am LRZ bestehenden Dokumentationsrichtlinien vorgenommen. Das Anfertigen eines forensischen Berichts wird für alle Sicherheitsvorfälle ab der Stufe „Mittel“ vorausgesetzt, ist allerdings bei allen Vorfällen sinnvoll.

Die Dokumentation muss Fakten sowie auch Schlussfolgerungen dokumentieren. Das folgende Kapitel führt forensische Richtilinien vor, an die sich jeder Bericht halten sollte. Des Weiteren stellt es eine Dokumentenvorlage vor, die schrittweise am Leibniz Rechenzentrum eingeführt werden soll.

Die Dokumentation muss so strukturiert sein, dass der Einstieg in das Materie zu einem späteren Zeitpunkt immer wieder möglich ist und sich sämtliche Untersuchungergebnisse nachvollziehen oder überprüfen lassen können. Dies setzt voraus, dass die Dokumentation übersichtlich und detailliert genug ist, um aufzuzeigen, welche Schritte unternommen wurden, welche Spuren entdeckt wurden sowie zu welchen Schlussfolgerungen die LRZ-Forensiker dadurch gekommen sind. Eines der grundlegenden forensischen Prinzipien ist die Wiederholbarkeit [4.2]. D.h. ein Bericht muss sicherstellen, dass bei Verwendung desselben Daten mit genau denselben Verfahren und Werkzeugen man erneut zu exakt identischen Ergebnissen gelangt. Da die Protokolle elektronisch gesichert werden, ist für die Vertraulichkeit sowie Integrität von Dokumenten ausreichend Sorge zu tragen.

Weiterhin gilt, dass das LRZ-CSIRT-Team (oder genauer gesagt der CSIRT-Hotliner) dafür Sorge tragen muss, dass die benötigten Informationen rechtzeitig eingepflegt werden und dass der Bericht nach dem Abschluss eine Security Incidents zeitnah angefertigt wird.

Zu einer solchen Dokumentation gehört eine Reihe von Kernpunkten. Die Dokumentation ist nach dem Makro-Mikro-Grundprinzip aufgebaut. Eine Dokumentenvorlage für Microsoft Word wird im Rahmen dieses Leitfaden zur Verfügung gestellt.

- **Kurze Zusammenfassung**: Dieser Abschnitt erläutert, aus welchen Gründen eine forensische Untersuchung der Computermedien notwendig war und listet die wichtigesten Ermittlungsergebnisse auf.
- **Beweisspuren**: Dieser Abschnitt liefert detallierte Informationen sowie Beschreibungen der sichergestellten Beweisspuren.
- **Untersuchungsbefunde**: In diesem Abschnitt werden die Ermittlungsergebnisse in der Reihenfolge der Wichtigkeit ihres Beweiswerts zusammenfassend aufgelistet. Vornehmlich geht es hierbei um die Beantwortung der Frage „Welche relevanten Spuren, die auf verdächtige Aktivitäten schließen lassen, während der Untersuchung entdeckt wurden?“.
- **Untersuchungsverlauf**: In diesem Abschnitt werden schließlich sämtliche Arbeitsschritte dargestellt, die zum Erreichen der Ermittlungsziele unternommen wurden. Hier sind Genauigkeit und Ausführlichkeit gefragt, damit ein andere Ermittler bei Bedarf nachvollziehen kann, was im Einzelnen getan wurde und die Befunde auch verifizieren kann. Darüber hinaus muss die Dokumentation so übersichtlich und detailliert sein, dass der ursprünglich ermittelnde Forensiker auch nach einem gewissen Zeitraum seine eigenen Analyse-Aufzeichnungen wieder zur Hand nehmen und seine Tätigkeiten überprüfen kann.

  Dieser Bereich sollte mit Hintergrundinformationen über die untersuchten Objekte (Daten aus der Live-Response-Phase, Arbeits- sowie Massenspeicherabbilder, usw.) beginnen. Anmerkungen zu jedem Schritt sollten kurz und sachlich gehalten werden. Zu jedem getätigten Untersuchungsschritt müssen sowohl die Beweggründe, die eingesetzten Hilfsmittel als auch die entdeckten Spuren aufgelistet werden. Es ist entscheidend, dass die Namen und Versionsnummern der verwendeten Tools festgehalten werden, da dies die Wiederholbarkeit und Verifizierung der Ergebnisse sicherstellt. Je nach Version des eingesetzten

Tools können, besonders nach größeren zwischenzeitlichen Updates, die Ergebnisse unterschiedlich ausfallen. Dies ist besonders wichtig, falls z.B. ein Antivirus-Scanner zur Überprüfung eines gemounteten Abbilds oder einiger exportierter Dateien eingesetzt wird.

Wurden Beweisspuren gesichert, so muss für eine lückenlose „Chain of Custody“ gesorgt werden. Dies setzt voraus, dass für jede sichergestellte Datei Prüfsummen berechnet und, zusammen mit Dateinamen, notiert wurden.

- **Weiterführende Informationen**: Dieser (optionaler) Abschnitt liefert Hintergrundinformationen über die Art und Weise, wie der Angriff ausgeführt wurde.
- **Ausblick**: Bei diesem (optionalen) Abschnitt handelt es sich um die Darstellung von möglichen weiteren Arbeitsschritten, die zu zusätzlichen, für die Ermittlung sachdienlichen Informationen (sowie unter Umständen auch zur Verifikation der bereits gesicherten Spuren) hätten führen können. Das Sicherstellen von möglichst vielen Beweismaterialien ist schließlich immer das Ziel einer forensischen Untersuchung. Da aber bei der Durchführung einer Analyse die Faktoren Zeit und Kosten in einem direkten Verhältnis zueinander stehen und der Aufwand einer Ermittlung mindestens proportional zu der Anzahl der durchzuführenden Schritte steigt, müssen gewisse Einschränkungen in Kauf genommen werden.

Der Bericht muss dafür Sorge tragen, dass die gesammelten Informationen nach dem Abschluss einer Untersuchung kausal und zeitlich korreliert werden. Der zeitliche Ablauf sollte nach Möglichkeit plausibel und nachvollziehbar erklärt werden. Nur wenn man in der Lage ist Zusammenhänge zwischen unterschiedlichen Teilbereichen herzustellen, kann man den kompletten Incident-Ablauf rekonstruieren und die Ergebnisse fundiert begründen. Des Weiteren ist der Rückgriff auf Drittquellen wie z.B. die Microsoft Knowledge Base ein wichtiger Schritt, um die Fundstücke/Ergebnisse der Analyse zu begründen. Ein Beispiel ist im Anhang D zu finden.

8. Fazit und Ausblick

Mit der Entscheidung Computer-Forensik Prozess einzuführen wird das Leibniz-Rechenzentrum den modernen Anforderung der IT-Landschaft gerecht. Bei der Erstellung von Richtlinien musste eine große Auswahl an derzeit erhältlichen Büchern, Leitfäden sowie Blogbeiträgen evaluiert und, bedingt durch die Prozessanforderungen am LRZ, auf die wesentlichen Maßnahmen reduziert werden. Im Verlauf dieses Leitfadens wurde eine große Reihe von möglichen forensischen Verfahren analysiert und solche ausgewählt, die eine erfolgreiche Umsetzung versprechen.

Mit der Umsetzung der in diesem Leitfaden ausgearbeiteten Richtlinien kommen auf das LRZ-CSIRT-Team sowie auf System-Administratoren neue organisatorische, prozessuale und technische Herausforderungen zu. Während die organisatorischen und prozessualen Aufgaben durch die Verknüpfung mit dem Incident-Response-Prozess weitestgehend abgedeckt werden, stellt die Werkzeugauswahl in der Praxis eine weitreichende Aufgabe dar.

Computerforenische Analysen mögen aufgrund der kostspieligen kommerziellen Softwarewerkzeuge finanziell oftmals unerschwinglich erscheinen. Derlei Programme sind jedoch für die Lösung von Sicherheitsvorfällen unter forensischen Gesichtspunkten am Leibniz-Rechenzentrum auch nicht zwingend erforderlich. Sie sind nunmal nichts anderes als Werkzeuge mit ihren jeweiligen Stärken und Schwächen - klassische Methoden der Computer-Forensik stoßen bei den heutigen Angriffen sowieso schnell an ihre Grenzen und die Spezifik der LRZ-Infrastruktur macht diese Werkzeugsammlungen nur bedingt für den Einsatz am Leibniz-Rechenzentrum geeignet. System- und Netzadministratoren am LRZ besitzen eine profunde Kenntnis der anstehenden Aufgaben und können entsprechend dafür erforderlichen Softwareassistenten mit Bedacht und Sachverstand auswählen - vorausgesetzt natürlich, sie wenden diese im Einklang mit forensischen Richtlinien an. Weiterhin wurde in der vorliegenden Arbeit für jedes Kapitel eine Reihe von kostenfreien Tools für die jeweiligen Aufgaben vorgestellt, deren Funktionen erläutert und deren praktische Anwendung auch teilweise beschrieben.

Generell wurden bei der Tool-Auswahl eine Reihe von Faktoren berücksichtigt:

- Usability,
- Funktionsumfang,
- Skalierbarkeit,
- Revisionssicherheit.

Grundsätzlich waren digitale Untersuchungen in der Vergangenheit vergleichsweise einfach. Selbst installierte Malware beeinträchtigte die Analyse eines befallenen Systems nur unwesentlich. Da sich die Aktivitäten eines Großteils der Schadsoftware leicht überwachen ließen, war es für Ermittler nur äußerst selten notwendig, kompromittierte IT-Systeme einer aufwendigen Untersuchung zu unterziehen.

Heutzutage sind Angreifer sehr versiert, wenn es darum geht, ihre Spuren zu verschleiern. Neue Angriffstechniken werden mit dem Ziel entwickelt, die Entdeckung durch Fachexperten zu vermeiden. Durch Einsatz von Techniken, die das Reverse Engineering vereiteln, auffälligen Netzwerktraffic durch Evasion-Techniken verbergen und lediglich minimale Spuren im Dateisystem hinterlassen, erschweren heutige Entwickler von Schadcode in zunehmendem Maße nicht nur jede Entdeckung, sondern auch jede herkömmliche Untersuchung. Dieser Trend begann ursprünglich mit LKM-Rootkits für Linux und führte schließlich zu einer Reihe ganz ähnlicher Verschleierungsmethoden auf Windows-Systemen.

Eine computerforensische Untersuchung besteht heutzutage niemals aus dem bloßen Anklicken von Buttons auf grafischen Benutzeroberflächen - das Zeitalter der spielerischen Forensik ist ohnehin vorbei. Sie besteht vielmehr aus der genauen Kenntnis aller Informations-Fragmente, denen man als Ermittler während einer

Untersuchung begegnet sowie aus einer logischen, sachverständigen und nachvollziehbaren Vorgehensweise beim Aufspüren, Sammeln und Interpretieren von Daten.

Diese Arbeit versucht ausgewählte, moderne Methoden der forensischen Analyse vorzustellen, damit man unter Zeitdruck mit passenden Maßnahmen auf eine Ausnahmesituation reagieren kann und dabei den Faktoren Infrastruktur, Zeit und Komplexität gerecht werden kann. Das A und O der Incident Response stellt dabei die neue Vorgehensweise der digitale Spurensuche direkt im Arbeitsspeicher eines Systems dar, wobei andere mögliche Maßnahmen ebenfalls bei Bedarf angewendet werden sollten.

Die Integration des Leitfadens in einen kontinuierlichen Verbesserungsprozess stellt eine grundlegende Voraussetzung dar für eine fortlaufende Aufdeckung von Optimierungspotentialen und neuen, fortgeschrittenen Verfahren. Änderungsbedarf ist mehr als notwendig und besteht im Rahmen der Fortentwicklung vor allem im Bereich der Abschaffung von noch vorhandenen Redundanzen. Die Mitarbeiter des LRZ sollten sich regelmäßig mit der Nutzung der Werkzeuge für eine erfolgreiche Beweissicherung vertraut machen, und die Ergebnisse der Übungen sollten rückgekopppelt werden. Dies ist notwendig, um aktuelle Bedrohungen zu erkennen und eine hohe Erfolgsrsate bei der Aufklärung von Security Incidents zu erreichen. Die rechtzeitige und nachvollziehbare Dokumentation aller Tätigkeiten und Erkenntnisse ist eine weitere Voraussetzung, die über Erfolg oder Misserfolg entscheidet.

A. Windows Live Response Toolkit richtig einsetzen

Im Rahmen dieses Leitfadens erstellte Windows Live Response Toolkit stellt grundlegende Funktionen zur Verfügung, um bei einem Sicherheitsvorfall flüchtige Daten von einem eingeschalteten Windows-System schnell und effizient zu sichern und dabei möglichst wenig Veränderungen an den Festplattendaten zu verursachen. Sobald eine stabile Version des Volatility Framework für Windows Server 2008-basierte Systeme zur Verfügung steht, wird das Toolkit größtenteils überflüssig. Dieser Abschnitt informiert, welche Schritte beim Einsatz zu beachten gilt - inklusive vorhandener Fallstricke.

1. Das Toolkit besteht aus einer Server- und einer Clientkomponente. Der Server wird auf der Forensikworkstation über die Batchdatei *start-server.bat* gestartet - nachdem die Fallnummer, der Name des Bearbeiters und der Port eingegeben wurden, wartet das Programm auf eingehende Verbindungen.
2. Anschließend kann auf der kompromittierten Maschine eine vertrauenswürdige Shell (*cmd.exe*) im Administrator-Modus aufgerufen und über die Datei *start.bat* der Live-Response-Skript gestartet werden. Hier gilt es die Ziel-IP Adresse sowie die Portnummer festzulegen, bevor der Datenaquisitionsvorgang initiiert werden kann.
3. Nun wird eine Verbindung zur Forensikworkstation hergestellt, und die gesammelten Daten werden übertragen und anschließend in einer Textdatei gespeichert.
4. Nach der Beendigung des Vorgangs muss der zuständige Forensiker die Integrität der erstellten Datei durch die Berechnung der Prüfsummen sicherstellen.

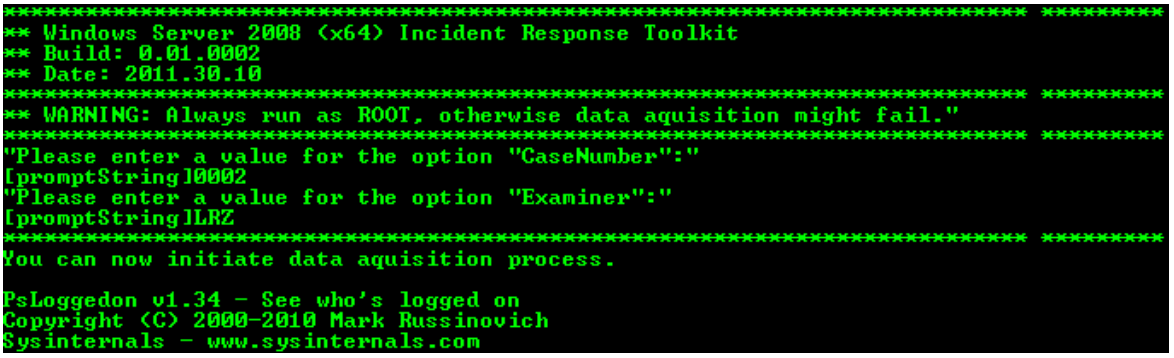

Abbildung A.1.: Windows Live Response Toolkit in Aktion.

Alternativ besteht die Möglichkeit die Datenaquisition lokal durchzuführen - hierfür wird die Datei *llr-lokal.bat* aufgerufen. Gesammelte Daten werden im neu erstellten Unterordner im Toolkitverzeichnis abgelegt - Prüfsummen werden automatisch berechnet. Dies ist ideal bei Maschinen, falls die lokale Netzkonfiguration eine direkte Verbindung unmöglich oder nur über Umwege möglich macht - Verwendung eines USB Sticks bei physischen oder eines virtuellen Datenträgers bei ESXi-Systemen wird vorausgesetzt.

Abhängig von der Server-Version und Patchlevel kann der Aufruf von *ipconfig* bzw. *nbtstat* versagen, in diesem Fall muss auf „lokale“ Dateien ausgewichen werden.

B. Linux Live Response Toolkit richtig einsetzen

Im Rahmen dieses Leitfadens verwendete Linux Live Response Toolkit basiert auf Forensik DVD aus iX special 10/2008 [Ewe] und stellt grundlegende Funktionen zur Verfügung, um bei einem Sicherheitsvorfall flüchtige Daten von einem eingeschalteten Linux-System schnell und effizient zu sichern. Dieser Abschnitt informiert, welche Schritte beim Einsatz zu beachten gilt - inklusive vorhandener Fallstricke.

1. Das Toolkit besteht aus einer Server- und einer Clientkomponente. Der Server wird auf der Forensikworkstation über die Batchdatei *start-server.bat* gestartet - nachdem die Fallnummer und der Port eingegeben wurden, wartet das Programm auf eingehende Verbindungen.

2. Anschließend kann auf der kompromittierten Maschine eine vertrauenswürdige Shell (*start.sh*) im Administrator-Modus aufgerufen und über die Datei *llr.sh %host% %port%* der Live-Response-Skript gestartet werden.

3. Im Anschluss gilt es Aufrufparameter festzulegen: standardmäßig sind bereits *checksum* (von jedem Aufruf wird eine Prüfsumme generiert) und *date* (vor jedem Aufruf wird ein Zeitstempel in die Ausgabedatei geschrieben). Wird *procmem* aktiviert, so wird der Prozessspeicher des Systems ausgelesen und ebenfalls abgespeichert - es ist zu erwarten, dass eine hohe Anzahl von Fehlermeldungen beim Ausführen generiert werden.

4. Nun wird eine Verbindung zur Forensikworkstation hergestellt, und die gesammelten Daten werden übertragen und anschließend in einer Textdatei gespeichert.

5. Nach der Beendigung des Vorgangs muss der zuständige Forensiker die Integrität der erstellten Datei durch die Berechnung der Prüfsummen sicherstellen.

6. Zum Schluss wird der Perl-Skript „llr-extract.pl“ aus dem Verzeichnis „tools“ aufgerufen - aus der Logdatei wird ein HTML-Bericht erzeugt, alle Memory Dumps und Binärdateien werden in ein separates Verzeichnis herausgeschrieben und im HTML-Bericht (Abbildung B.2) verlinkt.

```
=== Start of llr.sh Version 0.1 $Revision: 1.00 $

Available options:
  checksum  Checksum every external command with 'sha256sum -b' (which is
            also external) before running it.
  date      Run 'date' before every command to log exact start times. If
            not set, only start and end times of the script will be logged.
  procmem   Dump process memory. Takes a long time. May hang in rare cases
            if executed under X11.
  pcat      Use pcat (from The Coroners Toolkit) instead of pd (process dumper
            from trapkit.de)

Current options: >checksum date procmem<

Enter option to toggle or press enter to continue>

=== Start of llr.sh Version 0.1 $Revision: 1.00 $
===       CMD: ./llr.sh testrun
===      BASH: /home/tolwyn/llr/bin-x86-2.6/bash
===      PATH: /home/tolwyn/llr/bin-x86-2.6
=== SCRIPTDIR: ./
===   OPTIONS: checksum date procmem
===    OUTPUT: local file 'testrun'

Will write to logfile 'testrun'
Press Enter to start>
```

Abbildung B.1.: Linux Live Response Toolkit in Aktion.

Alternativ können Daten auch lokal auf einem in das FDateisystem eingehängten Datentraeger (USB-Stick bei physischen Systemen, virtueller Datenträger bei ESXi-Systemen) geschrieben werden - dies geschieht mit dem Aufruf *llr.sh %Ausgabedatei%*.

Eine Anmerkung zum Schluss: abhängig von der Kernelversion müssen die statisch gelinkten Binaries eventuell neu kompiliert werden. In der Testumgebung mit verschiedenen Kernelversionen ergaben sich allerdings keine Probleme.

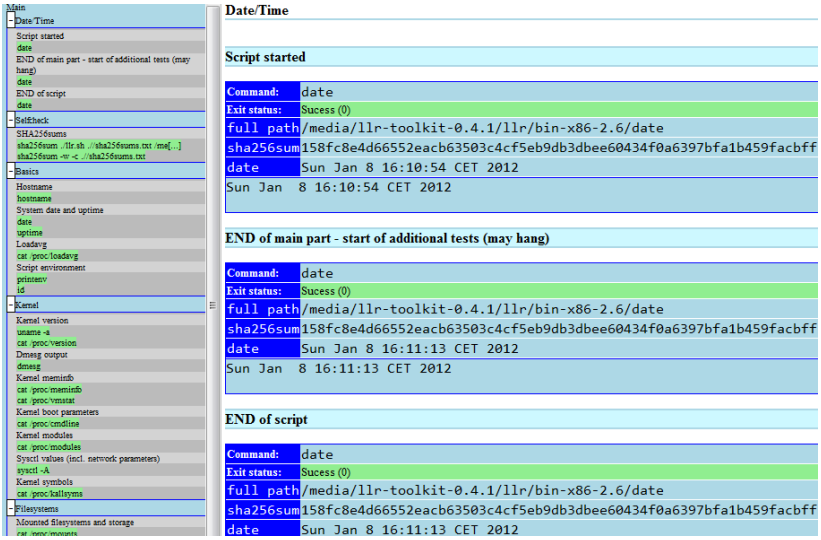

Abbildung B.2.: Mit Hilfe des Perl-Skripts „llr-extract.pl“ kann ein HTML-Bericht generiert .

C. Checklisten

Diese Checklisten sollen nicht dazu verleiten, eine Analyse Punkt für Punkt abzuarbeiten. Sie sollen vielmehr sicherstellen, dass alle oder zumindest die meisten der grundlegenden Arbeitsschritte durchgeführt wurden, und sie erleichtern sowohl Nachvollziehbarkeit als auch oft notwendige Wiederholung von einzelnen Schritten, ohne die Gefahr, dass etwas vergessen wird. Darüber hinaus wird eine systematische Vorgehensweise implementiert, die Voraussetzung dafür ist, den forensischen Prozess ständig zu optimieren.

Checkliste (Windows)

Mit dem Einsatz des Leitfadens sollen in der Regel folgende Zielsetzungen erreicht werden:

- Beweissammlung und –Sicherung sowie Auswertung digitaler Spuren,
- Quantifizieren des entstandenen Schadens,
- Identifizierung der verantwortlichen Sicherheitslücken,
- Identifikation des Angreifers,

Flüchtige Daten sicherstellen (falls das System noch eingeschaltet ist...):

- Flüchtigkeitsreihenfolge beachten,
- Die Praxis zeigt, dass die Erstellung eines Hauptspeicherabbilds für den Fortgang der Ermittlung von großer Wichtigkeit ist,
- Nach Möglichkeit das bereitgestellte Toolkit einsetzen,
- Bei der Sicherung von Spuren ist auf die Einhaltung der Beweismittelkette zu achten,
- Es muss jederzeit belegt werden können, dass die sichergestellten Spuren im Rahmen der Ermittlung nicht unbemerkt verändert wurden,
- Es sollte so weit wie möglich auf die Verwendung von systemeigenen Befehle verzichtet werden,
- Es sollten keine Programme oder Operationen ausgeführt werden, die die Zugriffszeiten vieler Dateien ändern.

Forensische Duplikation:

- Die Entscheidung über Durchführung einer forensischen Duplikation hängt von einigen Faktoren und der Priorität des Vorfalls ab,
- Bei der Durchführung ist an die vorher festgelegten Abläufe zu achten:
  - virtuelle Maschinen: Sicherung der vmdk-Datenträger durch die VMware-Administratoren,
  - physische Maschinen: Das System wird von einer bootfähigen Live-CD gebootet, mittels spezieller Software wird der komplette Inhalt des Datenträgers bitweise gesichert.

Forensische Analyse:

- Hauptspeicheranalyse:
  - Zum besseren Verständnis der Vorgänge am betroffenen System sollte immer das Hauptspeicherabbild analysiert werden. Voraussetzung hierfür ist natürlich, dass ein RAM-Dump angefertigt wurde.
  - Es spielt in der Regel keine Rolle, um welche Art von SI es sich handelt.
  - Bei der Analyse des Arbeitsspeicherinhalts können installierte Rootkits oder andere Arten von Malware entdeckt werden.
  - Die Analyse kann eventuell die Frage beantworten, wie der Angreifer auf das System gelangt ist. Dies gibt einen Anhaltspunkt, wonach man auf den anderen kompromittierten IT-Systemen suchen sollte.
- Registry-Analyse

- Setzt eine Kopie der Registry voraus (Hauptspeicherabbild, forensisches Duplikat oder Registry-Dump).
- Es spielt in der Regel keine Rolle, um welche Art von SI es sich handelt. Die Analyse der Registrierungsdatenbank kann auch im Rahmen der Hauptspeicheranalyse durchgeführt werden.
- Gewährt Einblicke in den möglichen zeitlichen Verlauf des Angriffs.
- Ein wesentliches Ergebnis der Registry-Analyse ist die Auswertung der Autostart-Einträge – falls ein Angreifer eine Hintertür oder andere schädliche Software auf dem System installiert hat, lässt sie sich evtl. auf diese Weise aufspüren.

☐ Logdatei-Analyse
- Falls kein forensisches Duplikat erstellt wurde, müssen Logdateien separat gespeichert werden.
- Die Analyse der Logdateien ist eines der wichtigsten Eckpfeiler, um Antworten auf die wesentlichen Fragen zu bekommen, nämlich „welcher Angriff durchgeführt wurde?“, „welche Schwachstelle ausgenutzt wurde?“ und „was hat der Angreifer auf dem System gemacht?“, und sollte daher immer angewendet werden.
- Gewährt Einblicke in den möglichen zeitlichen Verlauf des Angriffs.
- Unter Umständen lässt sich die Angriffsquelle bestimmen.
- Besteht der Verdacht, dass noch weitere Systeme in der Umgebung betroffen sind, können Protokolldateien Hinweise auf mögliche Kompromittierungen liefern.
- Auch finden sich hin und wieder Spuren, die den Angriffsmuster rekonstruieren lassen.
- Abhängig von der Art der Kompromittierung müssen evtl. mehrere Logdateien inspiziert werden.
- Folgende Informationen können für den Fortgang der Ermittlungen wichtig sein:
  - Hoher Ressourcenverbrauch,
  - Hohe Anzahl von gleichzeitig laufenden Prozesse,
  - Zu ungewöhnlichen Zeiten startende bzw. laufende Prozesse,
  - Mehrere Prozesse mit dem gleichen Namen,
  - Unbekannte Prozesse,
  - Abstürzende Prozesse,
  - Inaktive oder neue Benutzerkennung, die Prozesse starten,
  - Neue oder kürzlich reaktivierte Dienste (u.U. Indiz auf trojanisierte Prozesse od. Hintertüren)

☐ Erkennung der Rootkits
- Wird eine Malware auf dem System vermutet, sollte man frühzeitig daran denken, Gegenmaßnahmen einzuleiten.
- Ein forensisches Duplikat voraus, falls die Sicherung der Beweisintegrität im Vordergrund steht.

☐ Analyse des Dateisystems
- Bei einem Security Incident werden (fast) immer Spuren auf den Datenträgern hinterlassen.
- Wenn SIC dies für notwendig erachtet, sollten alle Möglichkeiten der Datenwiederherstellung in Betracht gezogen werden.
- Voraussetzung ist, dass ein forensisches Duplikat im Rahmen der Live-Response angefertigt wurde.
- Für den Fortgang der Ermittlungen kann der Inhalt des Papierkorbes und des freien Speicherplatzes von großer Bedeutung sein.

- Auch alle bekannten Möglichkeiten der Dateimaskierung sollten bedacht werden – vor allem bei Rootkits spielen ADS sowie unbenutzte Bereiche der Festplatte eine große Rolle.

- Timelime-Analyse
  - Wenn ein Angreifer auf einem System aktiv war, erweisen sich Timelines als ausgesprochen hilfreich, um seine Bewegungen zu verfolgen und zeitlichen Ablauf seiner Aktionen zu rekonstruieren.
  - Die Timeline-Analyse liefert Ansatzpunkte für weitere mögliche Fundorten von Spuren.
- Analyse der unbekannten Binärdateien
  - Stellenweise ist es notwendig sichergestellte binäre Dateien näher zu untersuchen.
  - Dies trifft vor allem dann zu, wenn Dateien in Verdacht stehen, schädlich zu sein, Antiviren-Lösungen allerdings keinen Alarm auslösen.
  - Die Art und Weise, wie sie geschrieben wurden, kann Erkenntnisse über ihre Funktionalität bringen.
  - Weiterhin kann die Analyse dazu dienen neue Angriffsmethoden frühzeitig zu identifizieren und entsprechend darauf zu reagieren.
  - Es ist außerdem denkbar, neue Erkenntnisse über den Täter zu gewinnen. Dies kann durchaus zur seiner Identifikation führen.

Checkliste (Linux)

Mit dem Einsatz des Leitfadens sollen in der Regel folgende Zielsetzungen erreicht werden:

- Beweissammlung und –Sicherung sowie Auswertung digitaler Spuren,
- Quantifizieren des entstandenen Schadens,
- Identifizierung der verantwortlichen Sicherheitslücken,
- Identifikation des Angreifers,

Flüchtige Daten sicherstellen (falls das System noch eingeschaltet ist...):

- Flüchtigkeitsreihenfolge beachten,
- Nach Möglichkeit das bereitgestellte Toolkit einsetzen,
- Bei der Sicherung von Spuren ist auf die Einhaltung der Beweismittelkette zu achten,
- Es muss jederzeit belegt werden können, dass die sichergestellten Spuren im Rahmen der Ermittlung nicht unbemerkt verändert wurden,
- Es sollte so weit wie möglich auf die Verwendung von systemeigenen Befehle verzichtet werden,
- Es sollten keine Programme oder Operationen ausgeführt werden, die die Zugriffszeiten vieler Dateien ändern.

Forensische Duplikation:

- Die Entscheidung über Durchführung einer forensischen Duplikation hängt von einigen Faktoren und der Priorität des Vorfalls ab,
- Bei der Durchführung ist an die vorher festgelegten Abläufe zu achten:
  - virtuelle Maschinen: Sicherung der vmdk-Datenträger durch die VMware-Administratoren,
  - physische Maschinen: Das System wird von einer bootfähigen Live-CD gebootet, mittels spezieller Software wird der komplette Inhalt des Datenträgers bitweise gesichert.

Forensische Analyse:

- Hauptspeicheranalyse:
  - Zum besseren Verständnis der Vorgänge am betroffenen System sollte immer das Hauptspeicherabbild analysiert werden. Voraussetzung hierfür ist natürlich, dass ein RAM-Dump angefertigt wurde und dass ein zuverlässiges Framework für die Analyse der Arbeitsspeicherabbilder zur Verfügung steht.
  - Es spielt in der Regel keine Rolle, um welche Art von SI es sich handelt.
  - Bei der Analyse des Arbeitsspeicherinhalts können installierte Rootkits oder andere Arten von Malware entdeckt werden.
  - Die Analyse kann eventuell die Frage beantworten, wie der Angreifer auf das System gelangt ist. Dies gibt einen Anhaltspunkt, wonach man auf den anderen kompromittierten IT-Systemen suchen sollte.
- Logdatei-Analyse
  - Falls kein forensisches Duplikat erstellt wurde, müssen Logdateien separat gespeichert werden.

- Die Analyse der Logdateien ist eines der wichtigsten Eckpfeiler, um Antworten auf die wesentlichen Fragen zu bekommen, nämlich „welcher Angriff durchgeführt wurde?“, „welche Schwachstelle ausgenutzt wurde?“ und „was hat der Angreifer auf dem System gemacht?“, und sollte daher immer angewendet werden.
- Gewährt Einblicke in den möglichen zeitlichen Verlauf des Angriffs.
- Unter Umständen lässt sich die Angriffsquelle bestimmen.
- Besteht der Verdacht, dass noch weitere Systeme in der Umgebung betroffen sind, können Protokolldateien Hinweise auf mögliche Kompromittierungen liefern.
- Auch finden sich hin und wieder Spuren, die den Angriffsmuster rekonstruieren lassen.
- Abhängig von der Art der Kompromittierung müssen evtl. mehrere Logdateien inspiziert werden.
- Folgende Informationen können für den Fortgang der Ermittlungen wichtig sein:
  - Hoher Ressourcenverbrauch,
  - Hohe Anzahl von gleichzeitig laufenden Prozesse,
  - Zu ungewöhnlichen Zeiten startende bzw. laufende Prozesse,
  - Mehrere Prozesse mit dem gleichen Namen,
  - Unbekannte Prozesse,
  - Abstürzende Prozesse,
  - Inaktive oder neue Benutzerkennung, die Prozesse starten,
  - Neue oder kürzlich reaktivierte Dienste (u.U. Indiz auf trojanisierte Prozesse od. Hintertüren)

- Aufspüren von nichtautorisierten Nutzer- oder Gruppenkonten
  - Für die Einschätzung des Angriffs ist es wichtig zu wissen, ob der Angreifer neue Nutzerkennung oder Gruppenkonten angelegt hat.
  - Wichtig ist dabei die Erkenntnis, dass dies ein recht unkomplizierter Vorgang ist, und daher meistens angewendet wird.
- Analyse der Konfigurationsdateien
  - Aus vielerlei Gründen ist es wichtig zu ermitteln, ob wichtige Konfigurationsdateien während des SIs modifiziert wurden, sei es für die Erstellung einer Hintertür oder Hinzufügen von neuen Diensten.
- Erkennung von „versteckten“ Dateien
  - Manchmal versuchen Angreifer ihre Spuren zu verwischen, indem sie bestimmte Dateien mit den gängigen Verfahren maskieren.
  - In diesem Abschnitt geht es darum mögliche Versteckorte aufzusuchen und versteckte Dateien zu entdecken.
  - Setzt ein forensisches Duplikat voraus.
- Erkennung der Rootkits
  - Wird eine Malware auf dem System vermutet, sollte man frühzeitig daran denken, Gegenmaßnahmen einzuleiten.
  - Setzt ein forensisches Duplikat voraus.
- Timelime-Analyse
  - Wenn ein Angreifer auf einem System aktiv war, erweisen sich Timelines als ausgesprochen hilfreich, um seine Bewegungen zu verfolgen und zeitlichen Ablauf seiner Aktionen zu rekonstruieren.
  - Die Timeline-Analyse liefert Ansatzpunkte für weitere mögliche Fundorten von Spuren.
- Analyse des proc-Dateisystems

- Setzt eine Kopie des Prozessdateisystems voraus (wird während der Live-Response-Phase erstellt).
- Enthält viele für forensische Untersuchungen interessante Informationen, wobei die zu suchenden Spuren von der konkreten Fragestellung des SIs abhängen.
- Ein wesentliches Ergebnis der Analyse des Prozessdateisystems ist die Auswertung der laufenden Prozesse – falls ein Angreifer eine Hintertür oder andere schädliche Software auf dem System installiert hat, lässt sie sich evtl. auf diese Weise aufspüren.

☐ Wiederherstellung von gelöschten Daten

- Bei einem Security Incident werden (fast) immer Spuren auf den Datenträgern hinterlassen.
- Wenn SIC dies für notwendig erachtet, sollten alle Möglichkeiten der Datenwiederherstellung in Betracht gezogen werden.
- Voraussetzung ist, dass ein forensisches Duplikat im Rahmen der Live-Response angefertigt wurde.

☐ Analyse der unbekannten Binärdateien

- Stellenweise ist es notwendig sichergestellte binäre Dateien näher zu untersuchen.
- Dies trifft vor allem dann zu, wenn Dateien in Verdacht stehen, schädlich zu sein, Antiviren-Lösungen allerdings keinen Alarm auslösen.
- Die Art und Weise, wie sie geschrieben wurden, kann Erkenntnisse über ihre Funktionalität bringen.
- Weiterhin kann die Analyse dazu dienen neue Angriffsmethoden frühzeitig zu identifizieren und entsprechend darauf zu reagieren.
- Es ist außerdem denkbar, neue Erkenntnisse über den Täter zu gewinnen. Dies kann durchaus zur seiner Identifikation führen.

D. Forensischer Bericht - Beispiel

Das nachfolgende Beispiel demonstriert wie alle Tätigkeiten sorgfältig dokumentiert und protokolliert werden können:

FORENSIK-BERICHT

1/7/2012

Fall-ID: 00001

Bearbeiter: Anton Romanyuk

Kurze Zusammenfassung

FALL-ID: 00001

Es wurde entdeckt, dass von der virtuellen Maschine CT103 ungewöhnlich viel Netz-Traffic verursacht wurde. Weiterhin stieg die Gesamtauslastung des Systems an. Ziel der Analyse war den Grund für das Fehlverhalten zu finden. Im Rahmen der Incident Response wurde ein Hauptspeicherabbild erstellt. Nach einer grundlegenden Auswertung des RAM-Dumps wurden Spuren des Darkness DDoS Bots entdeckt (eine Variante des Win32/Votwup.B).

Beweisspuren

FALL-ID: 00001

Dateiname	*Bearbeiter*	*Beschreibung*	*MD5-Hash*
ct103.vmem	Anton Romanyuk	Hauptspeicherabbild	5582970b95a6fdac 27ca070e83db970e
*executable.1464.ex_*	Anton Romanyuk	Verdächtige Binärdatei	4b47b5fe0e63c819 2b8b6aafa80c34cb
ct103_pstree.txt	Anton Romanyuk	Liste aller aktiven Prozesse	b14496eaa5ace4cf e75668c72f957a4b
ct103_psscan.txt	Anton Romanyuk	Liste aller aktiven & versteckten Prozesse	f68641ef061d859d c2e45992c1238f05
ct103_virustotal_1464.pdf	Anton Romanyuk	Ergebnis des Virusscan einer verdächtigen Datei	020b455ac9692adf a780949d6fea53ce
ct103_netscan.txt	Anton Romanyuk	Liste aller Netzverbindungen	13d243462fee2e07 486fc913cfd0d002
ct103_hivescan.txt	Anton Romanyuk	Liste der Registry-Hives	a0cb3eb2fd74e9e9f d33279520004b85
ct103_printkey.txt	Anton Romanyuk	Liste der aufgerufenen Registry-Schlüssel	92094d6af43b8075 c4a90b8705bf86c8
ct103_svcscan.txt	Anton Romanyuk	Liste der registrierten Dienste	3337158a9f0b66ff 79edcaaf4d4cac19
ct103_network.pcap	Anton Romanyuk	Mitschnitt des Netzverkehrs	bf0b6a5efbfd440f c7d50d220f95cc66

Untersuchungsbefunde

FALL-ID: 00001

- Untersuchungsergebnisse zeigen, dass eine Hintertür auf dem System aktiv ist.
- Weiterhin wurden Hinweise auf einen neuen Dienst *IpSectPro service new* entdeckt.
- Auswertung der Netzaktivität sowie der gesammelten Spuren zeigt, dass das System für DDoS Angriffe benutzt wurde.
- Installierte Malware wurde als Darkness DDoS Bot identifiziert[1].

[1] http://www.shadowserver.org/wiki/pmwiki.php/Calendar/20110123

Untersuchungsverlauf

FALL-ID: 00001

Zeit	*Aktion*	*Kommentar*
0:04	Snapshot erstellen... & md5-x64 ct103.vmem	Erstellung des Hauptspeicherabbilds mit „Snapshot erstellen…“, Berechnung der Prüfsumme mit MD5 1.1.18 (MD5-Hash: 5582970b95a6fdac27ca070e83db970e)
0:09	vol.py -f ct103.vmem pstree	Anzeige aller aktiven Prozesse in der Baumansicht (Volatility 2.1_alpha), unbekanntes Prozess *580846.exe* entdeckt (pid 1464), Ergebnis gespeichert in *ct103_pstree.txt* (MD5-Hash: b14496eaa5ace4cfe75668c72f957a4b)
0:11	vol.py -f ct103.vmem psscan	Auflistung aller aktiven, versteckten & beendeten Prozesse, bereits beendetes Prozess *darkness_8.exe* (Dropper?) entdeckt (pid 3560, exit time 2011-12-16 13:30:49), Ergebnis gespeichert in *ct103_psscan.txt* (MD5-Hash: f68641ef061d859dc2e45992c1238f05)
0:13	vol.py -f ct103.vmem procexedump -p 1464 -D dump/	Prozessdump zur Überprüfung mittels Virenscan, Ergebnis gespeichert in *executable.1464.ex_* (MD5-Hash: 4b47b5fe0e63c8192b8b6aafa80c34cb)
0:14	virustotal.com	Virusscan via virustotal.com, Datei identifiziert als *Backdoor:Win32/Votwup.B*, Ergebnis gespeichert in *ct103_virustotal_1464.pdf* (MD5-Hash: 020b455ac9692adfa780949d6fea53ce)
0:17	vol.py -f ct103.vmem netscan	Anzeige aller aktiven & beendeten Netzverbindungen, mehrere geschlossene Verbindungen zur bekannten C&C-IP entdeckt (*91.193.192.95:80*, pid 1464), geöffnetes Port 5005 entdeckt (pid 1464), Ergebnis gespeichert in *ct103_netscan.txt* (MD5-Hash: 13d243462fee2e07486fc913cfd0d002)
0:19	vol.py -f ct103.vmem hivelist	Auflistung aller Registry-Hives, HKCU-Key bei Memory-Offset *0x8be9b840* & HKLM bei *0x8b8889d0* lokalisiert, Ergebnis gespeichert in *ct103_hivescan.txt* (MD5-Hash: a0cb3eb2fd74e9e9fd33279520004b85)
0:25	vol.py ct103.vmem printkey -o ***offset*** -K ***string***	Typische Registryschlüssel werden schrittweise abgearbeitet, 0 Treffer in Autostart, 1 Treffer in Services (*IpSectPro service new*), Ergebnis gespeichert in *ct103_printkey.txt* (MD5-Hash:

		92094d6af43b8075c4a90b8705bf86c8)
0:32	vol.py svcscan	Überprüfung der eingetragenen Dienste, 1 Treffer (*lpSectPro service new*), Ergebnis gespeichert in *ct103_svcscan.txt* (MD5-Hash: 3337158a9f0b66ff79edcaaf4d4cac19)

Weiterführende Informationen

FALL-ID: 00001

Darkness DDoS Malware (auch bekannt als Votwup) ist für seine hohe Effizient bekannt: mit einer Handvoll infizierten Rechner können sogar größere Webseiten bzw. Server-Cluster mit fehlerhaften HTTP-Anfragen lahmlegen. Kompromittierte Systeme werden über mehrere C&C-Server im russischen Netzbereich kontrollieren und auf dem neuesten Stand gehalten[2].

Nach Angaben der Programmierer können 5000 Bots Server-Cluster, 15000-2000 beliebigen Server, unabhängig von verwendeten Schutzmaßnahmen in die Knie zwingen. Aktuelle Version der Malware ist 9H[3].

[2] http://krebsonsecurity.com/2011/08/digital-hit-men-for-hire/
[3] http://www.opensc.ws/unverified-items/14832-darkness-optima-ddos-bot.html

Ausblick

FALL-ID: 00001

Blockieren der einschlägigen C&C-Server IPs sowie Verwendung der aktuellen Virensignaturen sollte dafür Sorge tragen, dass die Verbreitung der Malware eingedämmt wird.

E. Verwendete Software

Name	Quelle
FTK Imager version 3.1.0	http://accessdata.com/support/adownloads
RegRipper 2.02	http://regripper.wordpress.com/regripper/
Volatility Framework 2.1_alpha	https://www.volatilesystems.com/default/volatility
The Sleuth Kit 3.2.3	http://www.sleuthkit.org/
Log Parser 2.2	http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24659
Log Parser Lizard GUI 2.0.0	http://www.lizard-labs.net/log_parser_lizard.aspx
HxD 1.7.7.0	http://mh-nexus.de/de/hxd/
Dependency Walker 2.2	http://www.dependencywalker.com/
ADS Spy 1.11	http://merijn.nu/programs.php
Cygwin 1.7.9-1	http://www.cygwin.com/
RootkitRevealer 1.71	http://technet.microsoft.com/de-de/sysinternals/bb897445
GMER 1.0.15.15641	http://www.gmer.net/
Notepad++ 5.9.8	http://notepad-plus-plus.org/
Python 2.6	http://www.python.org/getit/releases/2.6/
diStorm3 1.0	http://code.google.com/p/distorm/
PyCrypto 2.0.1	https://www.dlitz.net/software/pycrypto/
YARA 1.6	http://code.google.com/p/yara-project/

F. Source Code

```
@Echo Off
Color 0A
@echo ********************************************************************************* *********
@echo ** Windows Server 2008 (x64) Incident Response Toolkit
@echo ** Build: 0.01.0002
@echo ** Date: 2011.30.10
@echo ********************************************************************************* *********
@echo "Please enter a value for the option "CaseNumber":"
SET /P CASEN=[promptString]
@echo "Please enter a value for the option "Examiner":"
SET /P EXAM=[promptString]
@echo "Please enter a value for the option "Port":"
SET /P PORT=[promptString]
@echo ********************************************************************************* *********

:: variables declaration
set HEADER=CaseNumber: %CASEN%; Examiner: %EXAM%
set OUT=case_%CASEN%.txt

@echo You can now initiate data aquisition process.

@echo Do not forget to take the checksum of the resulting file!

@echo %HEADER% > %OUT%

netcat -l -vv -p %PORT% >> %OUT%
```

Listing F.1: Windows Toolkit start-server.bat

```
@Echo Off
Color 0A
@echo ********************************************************************************* *********
@echo ** Windows Server 2008 (x64) Incident Response Toolkit
@echo ** Build: 0.01.0002
@echo ** Date: 2011.30.10
@echo ********************************************************************************* *********
@echo ** WARNING: Always run as ROOT, otherwise data aquisition might fail."
@echo ********************************************************************************* *********
@echo "Please enter a value for the option "IP":"
SET /P IP=[promptString]
@echo "Please enter a value for the option "Port":"
SET /P PORT=[promptString]
@echo ********************************************************************************* *********

netcat -vv %IP% %PORT% -e llr.bat
```

Listing F.2: Windows Toolkit start.bat

```
@Echo Off
Color 0A
@echo ********************************************************************************* *********
@echo ** Windows Server 2008 (x64) Incident Response Toolkit
@echo ** Build: 0.01.0002
@echo ** Date: 2011.30.10
@echo ********************************************************************************* *********
@echo ** WARNING: Always run as ROOT, otherwise data aquisition might fail."
@echo ********************************************************************************* *********

@echo You can now initiate data aquisition process.

@echo ********************************************************************************* *********
@echo ** record system date and time
@echo ********************************************************************************* *********

@echo system date:
date /t
@echo system time:
time /t

@echo ********************************************************************************* *********
```

```
@echo ** record who is logged on the system
@echo ************************************************************************** *********

@echo Logged-on Users:
psloggedon.exe /accepteula

@echo ************************************************************************** *********

@echo Net Sessions:
net.exe session

@echo ************************************************************************** *********
@echo ** record information about network connections
@echo ************************************************************************** *********

@echo Cached NetBIOS Name Table:
nbtstat.exe -A

@echo ************************************************************************** *********

@echo Network Connections:
netstat.exe /ano

@echo ************************************************************************** *********
@echo ** determine all open files
@echo ************************************************************************** *********

@echo Open Files:
psfile.exe  /accepteula

@echo ************************************************************************** *********
@echo ** enumerate running processes
@echo ************************************************************************** *********

@echo Process Information:
tlist.exe /v

@echo ************************************************************************** *********

@echo Services:
tlist.exe /s

@echo ************************************************************************** *********

@echo Process Tree:
tlist.exe /t

@echo ************************************************************************** *********


@echo Process-to-Port Mapping:
Tcpvcon.exe /a /accepteula

@echo ************************************************************************** *********
@echo ** Detect ellicit sniffers
@echo ************************************************************************** *********

@echo Promiscuous Mode Check:
promiscdetect.exe /v

@echo  ************************************************************************** *********
@echo  ** Querry important configuration files
@echo  ************************************************************************** *********

@echo Network Configuration:
ipconfig.exe /all

@echo  ************************************************************************** *********
@echo  ** record system date and time
@echo  ************************************************************************** *********

@echo data aquisition finished at:
@echo system date: & date /t
@echo system time: & time /t

@echo ************************************************************************** *********
@echo Collection of volatile data completed...
@echo ************************************************************************** *********

:end
```

Listing F.3: Windows Toolkit llr.bat

```
@Echo Off
Color 0A
@echo ************************************************************************** *********
@echo ** Windows Server 2008 (x64) Incident Response Toolkit
@echo ** Build: 0.01.0002
@echo ** Date: 2011.30.10
@echo ************************************************************************** *********
@echo ** WARNING: Always run as ROOT, otherwise data aquisition might fail."
@echo ************************************************************************** *********
@echo "Please enter a value for the option "CaseNumber":"
SET /P CASEN=[promptString]
@echo "Please enter a value for the option "Examiner":"
SET /P EXAM=[promptString]
@echo "Please enter a value for the option "Destination":"
SET /P DEST=[promptString]
@echo ************************************************************************** *********

:: variables declaration
set SECTION=date
set HEADER=CaseNumber: %CASEN%; Examiner: %EXAM%
set OUT=case_%CASEN%_%SECTION%.txt
set FOLDER=%DEST%\case_%CASEN%

:: create case specific folders
mkdir %FOLDER%
mkdir %FOLDER%\md5

@echo You can now initiate data aquisition process.

:: ************************************************************************** *********
:: ** record system date and time
:: ************************************************************************** *********

@echo %HEADER% > %FOLDER%\%OUT%
@echo system date: >> %FOLDER%\%OUT% & date /t >> %FOLDER%\%OUT%
@echo system time: >> %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT%

:: ************************************************************************** *********
:: ** record who is logged on the system
:: ************************************************************************** *********

set SECTION=users
set OUT=case_%CASEN%_%SECTION%.txt

@echo time: > %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT% & echo %HEADER% >> %FOLDER%\%OUT%
@echo Logged-on Users: >> %FOLDER%\%OUT% & psloggedon.exe /accepteula >> %FOLDER%\%OUT%
@echo Net Sessions: >> %FOLDER%\%OUT% & net.exe session >> %FOLDER%\%OUT%

:: ************************************************************************** *********
:: ** record information about network connections
:: ************************************************************************** *********

set SECTION=network
set OUT=case_%CASEN%_%SECTION%.txt

@echo time: > %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT% & echo %HEADER% >> %FOLDER%\%OUT%
@echo Cached NetBIOS Name Table: >> %FOLDER%\%OUT% & nbtstat.exe -A >> %FOLDER%\%OUT%

:: ************************************************************************** *********

@echo Network Connections: >> %FOLDER%\%OUT% & netstat.exe /ano >> %FOLDER%\%OUT%

:: ************************************************************************** *********
:: ** determine all open files
:: ************************************************************************** *********

set SECTION=files
set OUT=case_%CASEN%_%SECTION%.txt

@echo time: > %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT% & echo %HEADER% >> %FOLDER%\%OUT%
@echo Open Files: >> %FOLDER%\%OUT% & psfile.exe /accepteula >> %FOLDER%\%OUT%

:: ************************************************************************** *********
:: ** enumerate running processes
:: ************************************************************************** *********

set SECTION=processes
set OUT=case_%CASEN%_%SECTION%.txt

@echo time: > %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT% & echo %HEADER% >> %FOLDER%\%OUT%

@echo Process Information: >> %FOLDER%\%OUT% & tlist.exe /v >> %FOLDER%\%OUT%
@echo: >> %FOLDER%\%OUT%
```

```
@echo Services: >> %FOLDER%\%OUT% & tlist.exe /s >> %FOLDER%\%OUT%
@echo: >> %FOLDER%\%OUT%
@echo Process Tree: >> %FOLDER%\%OUT% & tlist.exe /t >> %FOLDER%\%OUT%
@echo: >> %FOLDER%\%OUT%

:: ********************************************************************************* *********

@echo time: > %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT% & echo %HEADER% >> %FOLDER%\%OUT%
@echo Process-to-Port Mapping: >> %FOLDER%\%OUT% & Tcpvcon.exe /a /accepteula >> %FOLDER%\%OUT%

:: ********************************************************************************* *********
:: ** Detect ellicit sniffers
:: ********************************************************************************* *********

set SECTION=network
set OUT=case_%CASEN%_%SECTION%.txt

@echo time: >> %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT% & echo %HEADER% >> %FOLDER%\%OUT%
@echo Promiscuous Mode Check: >> %FOLDER%\%OUT% & promiscdetect.exe /v >> %FOLDER%\%OUT%
@echo: >> %FOLDER%\%OUT%

:: ********************************************************************************* *********
:: ** Querry important configuration files
:: ********************************************************************************* *********

@echo Network Configuration: >> %FOLDER%\%OUT% & ipconfig.exe /all >> %FOLDER%\%OUT%

:: ********************************************************************************* *********
:: ** record system date and time
:: ********************************************************************************* *********

set SECTION=date
set OUT=case_%CASEN%_%SECTION%.txt

@echo data aquisition finished at: >> %FOLDER%\%OUT%
@echo system date: >> %FOLDER%\%OUT% & date /t >> %FOLDER%\%OUT%
@echo system time: >> %FOLDER%\%OUT% & time /t >> %FOLDER%\%OUT%

@echo ********************************************************************************* *********
@echo Collection of volatile data completed...
@echo ********************************************************************************* *********
@echo Computing MD5...
@echo ********************************************************************************* *********

set SECTION=md5
set OUT=case_%CASEN%_%SECTION%.txt

:: md5sums will be stored in a separate folder, otherwise process will fail

MD5-x64 * > %FOLDER%\md5\%OUT%
MD5-x64 %FOLDER%\* >> %FOLDER%\md5\%OUT%

@echo ********************************************************************************* *********
@echo Finished! Press any key to continue...
@echo ********************************************************************************* *********

pause

:end
```

Listing F.4: Windows Toolkit llr-lokal.bat

```
#!./bin-x86-2.6/bash

enable shopt test [
shopt -s extglob
BASHDIR="${BASH%%/*([^/])}"
HDIR="${BASHDIR%%/*([^/])}"
SDIR="${BASHDIR##/*([^/])}"
SDIRNAME="${SDIR##*/}"
shopt -u extglob

export PATH="${BASHDIR}"
export HOME="${HDIR}"

echo "== BASH: ${BASH}"
echo "== HOME: ${HOME}"
echo "== PATH: ${PATH}"

if [ "${BASH%/*}" = "/bin" -o "${BASH%/*}" = "/usr/bin" ]; then
    echo "Please start script with static shell or from current dir, e.g.:"
    echo "# ./start-2.6.sh"
```

```
    exit 1
fi

PS1="${SDIRNAME}/bash \w \\\$ "
exec env -i PATH="${PATH}" HOME="${HOME}" PS1="${PS1}" TERM="${TERM}" bash --noprofile --
    rcfile "${HDIR}/.bashinit"
```

Listing F.5: Linux Toolkit start.sh

```
#!
#
# Linux Live Response script: start with static bash from trusted source
#
# Collect volatile data from a linux system.
# Designed to work with only statically linked binaries
# from a mounted external media.
#
# It is assumed that all binaries are in the same path as "bash" is.
#
# No files will be touched or read directly.
# Exceptions: /etc/passwd, utmp, wtmp (in the user mappings & login section)
#
# It is assumed that a disk dump is made directly afterwards to
# preserve this information.
#
# $Id: llr.sh,v 1.42 2008/08/08 08:58:21 enno Exp $
#
# Based upon iX IR Toolkit from Enno Ewers
#

VERSION="Version 0.1 \$Revision: 1.00 $"

# Paranoia - use builtins
enable cd break set unset true false test [ exit pwd shift shopt trap

#
# Fail if executed with system bash
# (This is not foolproof!)
#
if [ "${BASH%/*}" = "/bin" -o "${BASH%/*}" = "/usr/bin" -o \
     "${BASH%/*}" = "/sbin" -o "${BASH%/*}" = "/usr/sbin" -o \
     "${BASH%/*}" = "/usr/local/bin" -o "${BASH%/*}" = "/usr/local/sbin" ]; then
    echo "Please start script with static shell, e.g.:"
    echo "# <path-to-static-bins>/bash $0"
    exit 1
fi

#
# Tags
#
SECTION="===Section> "
SUBSEC="===Subsec> "
INFO="===Info> "
STARTCMD="===Cmd> "
EXITSTATUS="---Status> "

#
# Script options
#
declare -a allowed_options
declare -a options

allowed_options=(procmem checksum date pcat)
options=(checksum date)

#
# Function declarations
#
error() {
    echo "Error: $@" 1>&2
    echo "Error: $@"
    exit 1
}

usage() {
    echo "Usage: bash ./${0##*/} <host> <port>" 1>&2
    echo "   or: bash ./${0##*/} <outputfilename>" 1>&2
    exit 1
}

inf() {
    echo "I> ${CURSEC}: $*" 1>&2
```

```
    echo "${SUBSEC}$*"
}

log() {
    echo "$*"
}

loginfo() {
    echo "${INFO}$*"
}

section() {
    CURSEC="$*"
    echo "${SECTION}$*"
}

run() {
    BIN="$1"
    shift
    log "${STARTCMD}${BIN}" "$@"
    loginfo "full path: ${BINDIR}/${BIN}"
    if option checksum; then
    loginfo "sha256sum: "`"${BINDIR}/sha256sum" -b "${BINDIR}/${BIN}" | "${BINDIR}/cut" -d'
        ' -f1`
    fi
    if option date; then
    loginfo "date: "`"${BINDIR}/date"`
    fi
    { "${BINDIR}/${BIN}" "$@"; } 2>&1 | "${BINDIR}/tr" '\0' "${NULCHAR}"
    E=$PIPESTATUS
    log ""
    log "${EXITSTATUS}$E"
    if [ 0 -ne "$E" ]; then
    echo "==> Error running: ${BIN}" "$@" "(Exit code: $E)" 1>&2
    fi
}

encode_run() {
    FILENAME="$1"
    shift
    BIN="$1"
    shift
    log "${STARTCMD}${BIN}" "$@"
    loginfo "full path: ${BINDIR}/${BIN}"
    if option checksum; then
    loginfo "sha256sum: "`"${BINDIR}/sha256sum" -b "${BINDIR}/${BIN}" | "${BINDIR}/cut" -d'
        ' -f1`
    fi
    if option date; then
    loginfo "date: "`"${BINDIR}/date"`
    fi
    { stderrlog=`("${BINDIR}/${BIN}" "$@" | logenc "${FILENAME}"; exit ${PIPESTATUS}; )
        3>&1 1>&2 2>&3`; } 3>&1 1>&2 2>&3
    E=$?
    log "${stderrlog}"
    log "${EXITSTATUS}$E"
    if [ 0 -ne "$E" ]; then
    echo "==> Error running: ${BIN}" "$@" "(Exit code: $E)" 1>&2
    fi
}

logenc() {
    "${BINDIR}/uuencode" "$1"
}

toggleoption() {
    for sopt in $1; do
    for opt in "${allowed_options[@]}"; do
       if [ "x$sopt" = "x$opt" ]; then
        # valid option
        for xopt in "${options[@]}"; do
            if [ "x$sopt" = "x$xopt" ]; then
            # set - unset it
            options=(${options[@]%$sopt})
            echo "=> Removing $sopt"
            break 2;
            fi
        done
        # not set - set it
        options+=($sopt)
        echo "=> Adding $sopt"
       fi
```

```
    done
    done
}

option() {
    ret=1;
    for opt in "${options[@]}"; do
    if [ "x$1" = "x$opt" ]; then
        # valid option
        ret=0;
        break;
    fi
    done
    return $ret;
}



##############################
# Main part
##############################

SCRIPT="$0"
shopt -s extglob
SCRIPTDIR="${0%%*([^/])}"
SCRIPTNAME="${0##*/}"
BASHDIR="${BASH%%/*([^/])}"
shopt -u extglob
#BINDIR="${PWD}/${SCRIPTDIR}${STATICPATH}"
BINDIR="${BASHDIR}"
PATH=${BINDIR}

NULCHAR='\n'

TARGET=
REMOTE=
PORT=

#
# ARGV
#
if [ $# -eq 1 ]; then
    TARGET=$1
elif [ $# -eq 2 ]; then
    REMOTE=$1
    PORT=$2
    TARGET="/dev/tcp/${REMOTE}/${PORT}"
else
    usage
fi

if [ ! -x "${BINDIR}/bash" ]; then
    echo "Cannot find '${BINDIR}/bash'" 1>&2
    echo "Start script from its local directory (.../bash ./${SCRIPTNAME})" 1>&2
    exit 1
fi

ID=`"${BINDIR}/id" -u`

if [ "${ID}" -ne "0" ]; then
    echo "WARNING: Not running as root (ID=${ID})" 1>&2
    echo "         Most data acquisition will _fail_. " 1>&2
    echo "" 1>&2
    echo -n "Enter the letters \"ok\" to continue> " 1>&2
        read dummy
    if [ "$dummy" != "ok" ]; then
        echo "" 1>&2
        echo "Aborting. Please restart as root." 1>&2
        exit 1
    fi
fi

echo "=== Start of llr.sh ${VERSION}" 1>&2
echo "" 1>&2
echo "Available options:" 1>&2
echo "  checksum  Checksum every external command with 'sha256sum -b' (which is" 1>&2
echo "            also external) before running it." 1>&2
echo "  date      Run 'date' before every command to log exact start times. If" 1>&2
echo "            not set, only start and end times of the script will be logged." 1>&2
echo "  procmem   Dump process memory. Takes a long time. May hang in rare cases" 1>&2
echo "            if executed under X11." 1>&2
echo "  pcat      Use pcat (from The Coroners Toolkit) instead of pd (process dumper" 1>&2
echo "            from trapkit.de)" 1>&2
```

```
echo "" 1>&2
while true; do
    echo "Current options: >${options[*]}<" 1>&2
    echo "" 1>&2
    echo -n "Enter option to toggle or press enter to continue> " 1>&2
    read dummy
    if [ -z "$dummy" ]; then
    break;
    fi
    toggleoption "$dummy"
done
echo "" 1>&2

echo "=== Start of llr.sh ${VERSION}" 1>&2
echo "===       CMD: $0" "$@" 1>&2
echo "===      BASH: ${BASH}" 1>&2
echo "===      PATH: ${BINDIR}" 1>&2
echo "=== SCRIPTDIR: ${SCRIPTDIR}" 1>&2
echo "===   OPTIONS: ${options[@]}" 1>&2

if [ -n "${REMOTE}" ]; then
    echo "===    OUTPUT: remote ${TARGET}" 1>&2
else
    echo "===    OUTPUT: local file '${TARGET}'" 1>&2
fi
echo "" 1>&2

if [ -n "${REMOTE}" ]; then
    echo "Start TCP listener on target host $REMOTE port $PORT:" 1>&2
    echo "e.g.: # nc -l [-p] $PORT | tee logfile.out" 1>&2
else
    echo "Will write to logfile '${TARGET}'" 1>&2
fi
echo -n "Press Enter to start> "
read dummy
echo "" 1>&2

#
# audit script: kernel memory dump
#
audit_dump_mem() {
    # this will only log the first 1 Meg on some 2.6 kernels depending
    # on kernel configuration
    # check for "devmem_is_allowed" in /proc/kallsyms
    # this may hang the system under some circumstances
    inf "Memory dump allowed? (devmem_is_allowed should appear when yes)"
    run grep devmem /proc/kallsyms

    inf "Base memory dump"
    encode_run dev-mem.dump dd if=/dev/mem
}

#
# audit script: main
#

{
    loginfo "llr.sh: ${VERSION}"
    loginfo "Command invocation: $0" "$@"
    loginfo "bash: ${BASH}"
    loginfo "PATH: ${BINDIR}"
    loginfo "SCRIPTDIR: ${SCRIPTDIR}"
    loginfo "Options: ${options[@]}"
    loginfo "Output: ${TARGET}"

    section Date/Time
    inf "Script started"
    run date

    section Selfcheck
    inf SHA256sums

    if [ -f "${SCRIPTDIR}/sha256sums.txt" ]; then
    run sha256sum "$0" "${SCRIPTDIR}/sha256sums.txt" "${BINDIR}"/*
    run sha256sum -w -c "${SCRIPTDIR}/sha256sums.txt"
    else
    run sha256sum "$0" "${BINDIR}"/*
    fi

# ********************************************************************** *********
# ** record highly volatile information
# ********************************************************************** *********
```

```
    section Kernel

    inf "Dmesg output"
    run dmesg

    inf "Kernel meminfo"
    run cat /proc/meminfo
    run cat /proc/vmstat

    section Basics

    inf "Loadavg"
    run cat /proc/loadavg

# ********************************************************************************* *********
# ** record who is logged on the system
# ********************************************************************************* *********

    section Userinfo

    inf "User mappings and logins (reads /etc/passwd, utmp and wtmp)"
    run cat /etc/passwd
    run w
    run who
    run last

# ********************************************************************************* *********
# ** record information about network connections
# ********************************************************************************* *********

    section Network

    inf "List of applications and their PID associated with open network ports"
    run netstat -anp --inet
    run lsof -P -n -i -V

    inf "List of applications and their PID associated with open ports"
    run netstat -anp
    run lsof -P -n -V

# ********************************************************************************* *********
# ** record information about running processes
# ********************************************************************************* *********

    section Processes

    inf "Running processes overview"
    run ps -efl

# ********************************************************************************* *********
# ** record more information about network connections
# ********************************************************************************* *********

    section Network

    inf "Network interfaces"
    run netstat -i #ifconfig -s
    run netstat -aei #ifconfig -a

    inf "Routing and arp tables"
    run netstat -rn
    run arp -nv
    run cat /proc/net/arp


# ********************************************************************************* *********
# ** record basic information
# ********************************************************************************* *********

    section Basics

    inf "Hostname"
    run hostname

    inf "System date and uptime"
    run date
    run uptime

    inf "Script environment"
    run printenv
    run id

    section Kernel
```

```
    inf "Kernel version"
    run uname -a
    run cat /proc/version

    inf "Kernel boot parameters"
    run cat /proc/cmdline

    inf "Kernel modules"
    run cat /proc/modules

    inf "Sysctl values (incl. network parameters)"
    run sysctl -A

    inf "Kernel symbols"
    if [ -e /proc/ksyms ]; then
    run cat /proc/ksyms
    fi
    if [ -e /proc/kallsyms ]; then
    run cat /proc/kallsyms
    fi

    section Hardware

    inf "Hardware info and devices"
    run cat /proc/cpuinfo
    run cat /proc/interrupts
    run cat /proc/devices
    run cat /proc/ioports

# ********************************************************************************* *********
# ** record information about file system
# ********************************************************************************* *********

    section Filesystems

    inf "Mounted filesystems and storage"
    run cat /proc/mounts
    run cat /proc/partitions
    run cat /proc/swaps
    run df -h
    run df -k

# ********************************************************************************* *********
# ** dump process memory (long running)
# ********************************************************************************* *********

   section Processes
    inf "Running processes by /proc (some errors are expected)"
    for pid in /proc/[0-9]*; do
    run ls -nN  "${pid}"
    for file in "${pid}/"{cmdline,maps,status,stat,statm}; do
        run cat "${file}"
    done
    encode_run "${pid}-environ.raw" cat "${pid}/environ"
    run ls -n "${pid}"/fd
    done

    if option procmem; then
    inf "Process memory dump"
    for pid in /proc/[0-9]*; do
    npid="${pid##*/}"
    if [ "${npid}" -ne "$$" ]; then
        if option pcat; then
        # pcat
        encode_run "process-memory-${npid}.dump" pcat "${npid}"
        else
        # pd
        encode_run "process-memory-pid-${npid}.dump" pd_v1.1_lnx -p "${npid}"
        fi
    fi
    done
    fi

    section Date/Time
    inf "END of main part - start of additional tests (may hang)"
    run date

    section Additional Tests

    inf "Devices (may hang)"

    run ls -lR /dev
```

```
    run lspci -v

    for file in /proc/bus/*/devices; do
    run cat "${file}"
    done

    section Date/Time
    inf "END of script"
    run date

} >"${TARGET}"

echo "Done." 1>&2

if [ -n "${REMOTE}" ]; then
    echo "Do not forget to take the checksum on the remote host!" 1>&2
    echo "e.g.: # sha256sum logfile.log > logfile.log.sha256sum" 1>&2
else
    echo "Writing sha256sum to: ${TARGET}.sha256sum" 1>&2
    "${BINDIR}/sha256sum" "${TARGET}" | "${BINDIR}/tee" "${TARGET}.sha256sum" 1>&2
    echo "" 1>&2
fi
```

Listing F.6: Linux Toolkit llr.sh

Literaturverzeichnis

[A.] A., David: *David A's DWARF Page*. http://reality.sgiweb.org/davea/dwarf.html, Abruf: 24.12.2011

[Acc10] ACCESSDATA: *Forensic Toolkit Imager: Capture the Image, Preserve the Evidence*. AccessData Corp., 2010

[Aut] AUTY, Mike: *Error when processing windows 7 64-bit memory image*. http://code.google.com/p/volatility/issues/detail?id=79, Abruf: 30.11.2011

[Bas] BASSETTI, Nanni: *CAINE Live CD*. http://www.caine-live.net/, Abruf: 30.11.2011

[Bel] BELI, Emil: *Compiling kernel in openSUSE - easy way*. http://www.beli.ws/blog/?p=291, Abruf: 30.11.2011

[Boe] BOELEN, Michael: *Rootkit Hunter*. http://www.rootkit.nl/projects/rootkit_hunter.html, Abruf: 27.12.2011

[BSI11a] BSI: *Die Lage der IT-Sicherheit in Deutschland 2011*. Bundesamt für Sicherheit in Informationsmedien, 2011

[BSI11b] BSI: *Leitfaden 'IT-Forensik'*. Bundesamt für Sicherheit in Informationsmedien, 2011

[Buna] BUNDESDATENSCHUTZGESETZ: *Besondere Zweckbindung*. http://dejure.org/gesetze/BDSG/31.html, Abruf: 10.10.2011

[Bunb] BUNDESDATENSCHUTZGESETZ: *Datenvermeidung und Datensparsamkeit*. http://dejure.org/gesetze/BDSG/3a.html, Abruf: 10.10.2011

[BW] BW: *Voice over IP - Internet-Telefonie verliert Anhänger*. http://www.business-wissen.de/unternehmensfuehrung/voice-over-ip-internet-telefonie-verliert-anhaenger/, Abruf: 06.08.2011

[Cara] CARRIER, Brian: *Mac-robber*. http://wiki.sleuthkit.org/index.php?title=Mac-robber, Abruf: 06.01.2012

[Carb] CARRIER, Brian: *The Sleuth Kit*. http://sleuthkit.org/, Abruf: 24.12.2011

[Carc] CARRIER, Brian: *Timelines*. http://wiki.sleuthkit.org/index.php?title=Timelines, Abruf: 24.12.2011

[Car05] CARRIER, Brian: *File System Forensic Analysis*. Addison-Wesley Longman, 2005

[Car09] CARVEY, Harlan: *Windows Forensic Analysis DVD Toolkit, Second Edition*. Syngress, 2009

[Cas] CASE, Andrew: *Linux Memory Analysis Workshop*. https://media.blackhat.com/bh-us-11/Case/BH_US_11_Case_Linux_Slides.pdf, Abruf: 24.12.2011

[CB] CB: *Seagate bringt weltweit erste 4-TB-Festplatte*. http://www.computerbase.de/news/2011-09/seagate-bringt-weltweit-erste-4-tb-festplatte/, Abruf: 10.09.2011

[CER] CERT, DFN: *Suche nach Hinweisen auf Kompromittierung am Beispiel von UNIX / Linux*. http://www.dfn-cert.de/informationen/themen/incident-response-informationen/nachsehen-linux.html, Abruf: 30.11.2011

[CM08] CAMERON MALIN, James A. Eoghan Casey C. Eoghan Casey: *Malware Forensics: Investigating and Analyzing Malicious Code*. Syngress, 2008

[CP03] CHRIS PROSISE, Matt P. Kevin Mandia M. Kevin Mandia: *Incident Response and Computer Forensics, Second Edition*. McGraw-Hill/Osborne, 2003

[CP08] CHRIS POGUE, Todd H. Cory Altheide A. Cory Altheide: *UNIX and Linux Forensic Analysis DVD Toolkit*. Syngress, 2008

[Crea] CREMER, Florian: *DLL-Injektion über API-Hooking - Eine Einführung in das Integrieren eigenen Codes in fremde Anwendungen*. http://phpconference.com/itr/online_artikel/psecom,id,434,nodeid,56.html, Abruf: 24.12.2011

[Creb] CREMER, Florian: *DLL-Injektion über API-Hooking: Eine Einführung in das Integrieren eigenen Codes in fremde Anwendungen*. http://entwickler.com/itr/online_artikel/psecom,id,434,nodeid,56.html, Abruf: 16.09.2011

[Data] DATA, Access: *Forensic Toolkit Imager: Capture the Image, Preserve the Evidence*. http://accessdata.com/downloads/current_releases/imager/FTKImager_ReleaseNotes_3-0-1.pdf, Abruf: 30.11.2011

[Datb] DATA, ASR: *SmartMount*. http://www.asrdata.com/forensic-software/smartmount/, Abruf: 30.11.2011

[Datc] DATA, Get: *Mount Image Pro v4 - Forensic Software*. http://www.mountimage.com/, Abruf: 30.11.2011

[DFR] DFRWS: *DFRWS 2005 Forensics Challenge*. http://www.dfrws.org/2005/challenge/, Abruf: 30.11.2011

[dig] DIGFOR: *Accessing VMFS partitions*. http://digfor.blogspot.com/2011/04/accessing-vmfs-partitions.html, Abruf: 10.11.2011

[DiMa] DIMINO, Andre M.: *BlackEnergy competitor - The Darkness DDoS Bot*. http://www.shadowserver.org/wiki/pmwiki.php/Calendar/20101205, Abruf: 30.11.2011

[DiMb] DIMINO, Andre M.: *Using 'volatility' to study the CVE-2011-0611 Adobe Flash 0-day*. http://sempersecurus.blogspot.com/2011/04/using-volatility-to-study-cve-2011-6011.html, Abruf: 10.11.2011

[dis] DISTORM: *distorm - Powerful Disassembler Library For x86/AMD64*. http://code.google.com/p/distorm/, Abruf: 30.11.2011

[eHo] EHOW: *How to Dump Linux Memory*. http://www.ehow.co.uk/how_6386134_dump-linux-memory.html, Abruf: 20.11.2011

[Ewe] EWERS, Enno: *Forensik DVD aus iX special 10/2008*. http://computer-forensik.org/tools/ix/ix-special/, Abruf: 28.12.2011

[EYV] EUGENE Y. VASSERMAN1, Nicholas H.: *SILENTKNOCK: Practical, Provably Undetectable Authentication*. http://www.cs.umn.edu/~hopper/silentknock_esorics.pdf, Abruf: 30.11.2011

[fen] FENSE e: *Helix3 Pro*. http://www.e-fense.com/helix3pro.php, Abruf: 10.10.2011

[Flo] FLOYD, Paul J.: *ObjectViewer, an nm Front End*. http://paulf.free.fr/objectviewer.html, Abruf: 29.12.2011

[for] FORENSICSWIKI: *Write Blockers*. http://www.forensicswiki.org/wiki/Write_Blockers, Abruf: 10.10.2011

[Fos10] FOSSI, Marc: *Symantec Internet Security Threat Report: Trends for 2010*. Symantec Corporation, 2010

[Fär11] FÄRBINGER, Peter M.: *E-3*. B4Bmedia.net, 2011

[Fra] FRATEPIETRO, Stefano: *DEFT Linux*. http://www.deftlinux.net/, Abruf: 30.11.2011

[FS] F-SECURE: *Blacklight*. http://www.f-secure.com/en/web/labs_global/removal/blacklight, Abruf: 24.12.2011

[Ges10] GESCHONNECK, Alexander: *Computer-Forensik: Computerstraftaten erkennen, ermitteln, aufklären, 4., aktualisierte Auflage*. dpunkt Verlag, 2010

[GF09] GERTI FOEST, Torsten V.: *Linux Kernelarchitektur: Konzepte, Strukturen und Algorithmen von Kernel 2.6*. DFN-Verein, 2009

[GH05] GREG HOGLUND, James B.: *Rootkits. Das Standardwerk zu Funktionsweise, Entwicklung und Entdeckung von Rootkits für Windows 2000/XP.* Addison Wesley Verlag, 2005

[Gil] GILBERT, Nzeka: *Rootkits unter Windows-Plattformen*. http://remoteshell-security.com/dokumente/rootkits.pdf, Abruf: 30.11.2011

[Gle] GLEASON, BJ: *Alternatives to Helix3*. http://www.forensicfocus.com/alternatives-to-helix3, Abruf: 10.11.2011

[Gmb] GMBH, Plan42: *OSSIM*. http://www.plan42.com/index.php/de/open-source-business-solutions/ossim-security-management/ossim-details, Abruf: 22.08.2011

[GME] GMER: *GMER - all your rootkits are belong to us*. http://www.gmer.net/, Abruf: 24.12.2011

[Goo] GOOGLE: *Advanced Operators*. http://www.google.de/help/operators.html, Abruf: 12.08.2011

[Gru] GRUNDGESETZ: *Recht auf informationelle Selbstbestimmung*. http://dejure.org/gesetze/GG/2.html, Abruf: 10.10.2011

[Hala] HALE, Michael: *Example usage cases and output for Volatility commands*. http://code.google.com/p/volatility/wiki/CommandReference, Abruf: 30.11.2011

[Halb] HALE, Michael: *Full Dev Installation for Volatility 2.0*. http://code.google.com/p/volatility/wiki/FullInstallation, Abruf: 30.11.2011

[Halc] HALE, Michael: *malwarecookbook*. http://code.google.com/p/malwarecookbook/, Abruf: 24.12.2011

[Hara] HARBOUR, Nicholas: *dcfldd*. http://dcfldd.sourceforge.net/, Abruf: 30.11.2011

[Harb] HARLEY, David: *TDL4 rebooted*. http://blog.eset.com/2011/10/18/tdl4-rebooted, Abruf: 24.12.2011

[HBG] HBGARY: *Responder Professional*. http://hbgary.com/responder-pro, Abruf: 30.11.2011

[Hey] HEYSOFT: *LADS - List Alternate Data Streams*. http://www.heysoft.de/de/software/lads.php, Abruf: 24.12.2011

[HR10] HELMUT REISER, Wolfgang H. Stefan Metzger M. Stefan Metzger: *Integriertes Management von Sicherheitsvorfällen*. LRZ, 2010

[IDC] IDC: *Insider Risk Management: A Framework Approach to Internal Security*. http://www.rsa.com/solutions/business/insider_risk/wp/10388_219105.pdf, Abruf: 07.10.2011

[inu] INUXWEBLOG: *Image Your Hard Drive using dd*. http://www.linuxweblog.com/dd-image, Abruf: 10.10.2011

[Jon] JONES, M. T.: *Anatomy of the Linux slab allocator*. http://www.ibm.com/developerworks/linux/library/l-linux-slab-allocator/, Abruf: 20.11.2011

[JS09] JORGEN SCHÄFER, Marcus P.: *DFN Mitteilungen Ausgabe 76*. DFN-Verein, 2009

[Lab] LABS, Lizard: *Log Parser Lizard GUI - FREE Query Software*. http://www.lizard-labs.net/log_parser_lizard.aspx, Abruf: 24.12.2011

[Lei] LEITEL, Marcel: *Netflow - der unterschätzte Freund*. http://www.dmt-service.de/fileadmin/dmt/Downloads/Netflow-der_unterschaetzte_Freund.pdf, Abruf: 22.08.2011

[Liga] LIGH, Michael: *32-bit and 64-bit Application Data in the Registry*. http://msdn.microsoft.com/en-us/library/windows/desktop/ms724072(v=vs.85).aspx, Abruf: 30.11.2011

[Ligb] LIGH, Michael: *Volatility's New Netscan Module*. http://mnin.blogspot.com/2011/03/volatilitys-new-netscan-module.html, Abruf: 30.11.2011

[Lit] LITZENBERGER, Dwayne C.: *PyCrypto - The Python Cryptography Toolkit*. https://www.dlitz.net/software/pycrypto/, Abruf: 30.11.2011

[LRZa] LRZ: *IP-Adressumsetzung (NAT) als Ersatz für Proxyserver*. http://www.lrz.de/services/netzdienste/nat-o-mat/, Abruf: 20.08.2011

[LRZb] LRZ: *Das LRZ in Kürze*. http://www.lrz.de/wir/lrz-flyer/de/, Abruf: 06.08.2011

[Lyo] LYON, Gordon: *Nmap*. http://nmap.org/, Abruf: 30.11.2011

[Mac] MACHOR, Mitchell: *The Forensic Analysis of the Microsoft Windows Vista Recycle Bin*. http://www.forensicfocus.com/downloads/forensic-analysis-vista-recycle-bin.pdf, Abruf: 24.12.2011

[Mad] MADIANT: *Memoryze*. http://www.mandiant.com/products/free_software/memoryze/, Abruf: 30.11.2011

[Mau03] MAUERER, Wolfgang: *DFN Mitteilungen Ausgabe 77*. Carl Hanser Verlag GmbH & CO. KG, 2003

[McD] MCDONAGH, Steve: *The tao of SQL Injection exploits*. http://dominoyesmaybe.blogspot.com/2008/05/tao-of-sql-injection-exploits.html, Abruf: 24.12.2011

[mch] MCHOUTEAU: *SQL queries against a variety of log files and other system data sources*. http://visuallogparser.codeplex.com/, Abruf: 24.12.2011

[Mer] MERIJN: *ADS Spy*. http://merijn.nu/programs.php, Abruf: 24.12.2011

[Met11] METZGER, Stefan: *Informations Security Incident Management*. LRZ, 2011

[MICa] M. I. COHEN, A. W. D. J. Collett C. D. J. Collett: *Digital Forensics Research Workshop 2008 - Submission for Forensic Challenge*. http://sandbox.dfrws.org/2008/Cohen_Collet_Walters/Digital_Forensics_Research_Workshop_2.pdf, Abruf: 24.12.2011

[Micb] MICROSOFT: *Download and Install Debugging Tools for Windows*. http://msdn.microsoft.com/en-us/windows/hardware/gg463009.aspx, Abruf: 10.09.2011

[Micc] MICROSOFT: *Microsoft Security Bulletins*. http://technet.microsoft.com/en-us/security/bulletin/, Abruf: 24.12.2011

[Micd] MICROSOFT: *TrojanSpy:Win32/Laqma.B*. http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=TrojanSpy%3aWin32%2fLaqma.B, Abruf: 30.11.2011

[Mil] MILLER, Steve P.: *Dependency Walker 2.2*. http://www.dependencywalker.com/, Abruf: 24.12.2011

[MJWB03] MOIRA J. WEST-BROWN, Klaus-Peter K. Don Stikvoort S. Don Stikvoort: *Handbook for Computer Security Incident Response Teams (CSIRTs)*. DFN-Verein, 2003

[MS11] MICHAEL SOLOMON, Neil B. Ed Tittel T. Ed Tittel: *Computer Forensics JumpStart*. John Wiley & Sons, 2011

[MTC] MTC: *Most Prolific BotNet Command and Control Servers and Filters*. http://mtc.sri.com/live_data/cc_servers/, Abruf: 10.11.2011

[Mul] MULLER, Lance: *New version of EnCase includes stand-alone utility to capture RAM*. http://www.forensickb.com/2008/06/new-version-of-encase-includes-stand.html, Abruf: 10.09.2011. http://www.forensickb.com/2008/06/new-version-of-encase-includes-stand.html

[Mum] MUMMERT, Steria: *Telekommunikation: Sicherheit genießt bei IT-Investitionen höchste Priorität*. http://www.steria-mummert.de/presse/presseinformationen/telekommunikation-sicherheit-geniesst-bei-it-investitionen%-hoechste-prioritaet, Abruf: 06.08.2011

[Mur] MURILO, Nelson: *locally checks for signs of a rootkit*. http://www.chkrootkit.org/, Abruf: 24.12.2011

[NfS] NFSEN: *NfSen - Netflow Sensor*. http://nfsen.sourceforge.net/, Abruf: 20.08.2011

[Pan10] PANDALABS: *Annual Report PandaLabs 2010*. PandaLabs, 2010

[pyf] PYFLAG: *RAID Reassembly - A forensic Challenge*. http://pyflag.sourceforge.net/Documentation/articles/raid/reconstruction.html, Abruf: 10.11.2011

[Pyt] PYTHON: *Python Programming Language*. http://python.org/, Abruf: 30.11.2011

[RFCa] *RFC 3227 - Guidelines for Evidence Collection and Archiving*. http://www.faqs.org/rfcs/rfc3227.html),author={RFC},urldate={10.09.2011}, Abruf: 10.09.2011

[RFCb] RFC: *Rate Limiting*. http://freesoft.org/CIE/RFC/1812/74.htm, Abruf: 12.08.2011

[RFCc] RFC: *Transmission Control Protocol*. http://tools.ietf.org/html/rfc793, Abruf: 12.08.2011

[Ric] RICHARD, Golden G.: *Bringing Linux Support to Volatility*. http://dfsforensics.blogspot.com/2011/03/bringing-linux-support-to-volatility.html, Abruf: 24.12.2011

[Rusa] RUSSINOVICH, Mark: *Inside the Windows Vista Kernel: Part 2*. http://technet.microsoft.com/en-us/magazine/2007.03.vistakernel.aspx, Abruf: 10.09.2011

[Rusb] RUSSINOVICH, Mark: *RootkitRevealer 1.71*. http://technet.microsoft.com/de-de/sysinternals/bb897445, Abruf: 24.12.2011

[SA07] STEVEN ANSON, Steve B.: *Mastering Windows Network Forensics and Investigation*. Wiley Publishing, Inc., 2007

[Sch] SCHONSCHEK, Oliver: *Gefahr durch Insider: So sehen Innentäter aus*. http://www.datenschutz-praxis.de/fachwissen/fachartikel/gefahr-durch-insider-so-sehen-innentater-aus/, Abruf: 10.11.2011

[SG] STEVE GIBSON, Nanni B.: *AIR - Automated Image and Restore*. http://sourceforge.net/apps/mediawiki/air-imager/, Abruf: 30.11.2011

[Sha08] SHAVERS, Brett: *A Discussion of Virtual Machines Related to Forensics Analysis*. Virtual Forensics, 2008

[Sofa] SOFTWARE, Cyber-Ark: *Trust, Security and Passwords*. http://www.cyber-ark.com/resources/by-resource-type/white-papers.asp, Abruf: 07.10.2011

[Sofb] SOFTWARE, Guidance: *Guidance Software*. http://www.guidancesoftware.com/, Abruf: 10.10.2011

[Sofc] SOFTWARE, Guidance: *Guidance Software*. http://www.guidancesoftware.com/, Abruf: 10.10.2011

[Sof06] SOFTWARE, Guidance: *EnCase Forensic: Detailed Product Description*. Guidance Software, 2006

[Sop] SOPHOS: *Sophos Anti-Rootkit - Verbergen sich Rootkits auf Ihrem Computer?* http://www.sophos.com/de-de/products/free-tools/sophos-anti-rootkit.aspx, Abruf: 24.12.2011

[Sui] SUICHE, Matthieu: *MoonSols Windows Memory Toolkit*. http://www.moonsols.com/windows-memory-toolkit/, Abruf: 10.09.2011

[Supa] SUPPORT, Microsoft: *Description of how the Attachment Manager works in Microsoft Windows*. http://support.microsoft.com/kb/883260/en-us, Abruf: 24.12.2011

[Supb] SUPPORT, Microsoft: *Log Parser 2.2*. http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=24659, Abruf: 24.12.2011

[Supc] SUPPORT, Microsoft: *Warnung vor dem Wurmvirus Win32/Conficker*. http://support.microsoft.com/kb/962007/de, Abruf: 24.12.2011

[Sym] SYMANTEC: *Backdoor.Maxload*. http://www.symantec.com/security_response/writeup.jsp?docid=2004-110420-4659-99l, Abruf: 30.11.2011

[Sysa] SYSINTERNALS: *PsLoggedOn v1.34*. http://technet.microsoft.com/en-en/sysinternals/bb897545, Abruf: 16.09.2011

[Sysb] SYSTEMS, Volatile: *Linux Memory Forensics*. http://volatilesystems.blogspot.com/2008/07/linux-memory-analysis-one-of-major.html, Abruf: 20.11.2011

[Tab] TABLEAU: *Tableau TD1 Forensic Duplicator*. http://www.tableau.com/index.php?pageid=products&model=TD1, Abruf: 10.11.2011

[Tor] TORTOISESVN: *TortoiseSVN - the coolest interface to (Sub)version control*. http://tortoisesvn.net/, Abruf: 30.11.2011

[Uni] UNIVERSITY, Carnegie M.: *2011 CyberSecurity Watch Survey: How Bad Is the Insider Threat?* http://www.cert.org/archive/pdf/CyberSecuritySurvey2011Data.pdf, Abruf: 07.10.2011

[VB] VB: *BSI kritisiert Notfallpläne*. http://www.virusbeseitigung.de/news/bsi-kritisiert-notfallplaene, Abruf: 06.08.2011

[Ver] VERFASSUNGSSCHUTZ, Bundesamt für: *Sicherheitslücke Mensch - Der Innentäter als größte Bedrohung für die Unternehmen*. http://www.verfassungsschutz.de/de/publikationen/spionageabwehr_geheimschutz/faltblatt_ws_1008_2_sicherheitsluecke_mensch/, Abruf: 07.10.2011

[VMw] VMWARE: *VDDK Documentation*. http://www.vmware.com/support/developer/vddk/, Abruf: 30.11.2011

[Wat09] WATKINS, Kevin: *VoIP Vulnerabilities*. McAfee, Inc., 2009

[Wika] WIKI, Forensics: *Linux Memory Analysis*. http://www.forensicswiki.org/wiki/Linux_Memory_Analysis, Abruf: 20.11.2011

[Wikb] WIKI, Forensics: *List of Volatility Plugins*. http://www.forensicswiki.org/wiki/List_of_Volatility_Plugins, Abruf: 30.11.2011

[Wikc] WIKI, Forensics: *Tools: Memory Imaging*. http://www.forensicswiki.org/wiki/Tools:Memory_Imaging, Abruf: 20.11.2011

[Wikd] WIKIPEDIA: *Desktop Window Manager*. http://en.wikipedia.org/wiki/Desktop_Window_Manager, Abruf: 30.11.2011

[Wike] WIKIPEDIA: *Locard'sche Regel*. http://de.wikipedia.org/wiki/Locard'sche_Regel, Abruf: 10.09.2011

[Wikf] WIKIPEDIA: *Port scanner*. http://en.wikipedia.org/wiki/Port_scanner, Abruf: 12.08.2011

[Wikg] WIKIPEDIA: *Rootkit*. http://en.wikipedia.org/wiki/Rootkit, Abruf: 05.01.2012

[Wikh] WIKIPEDIA: *Windows Registry*. http://en.wikipedia.org/wiki/Windows_Registry, Abruf: 30.11.2011

[Yar] YARA: *yara-project - A malware identification and classification tool*. http://code.google.com/p/yara-project/, Abruf: 30.11.2011

[zai] ZAIRON: *Vmware snapshot and SSDT*. http://zairon.wordpress.com/2008/06/04/vmware-snapshot-and-ssdt/, Abruf: 10.11.2011

[Zbi] ZBIKOWSKI, Mark: *Microsoft PE and COFF Specification*. http://msdn.microsoft.com/library/windows/hardware/gg463125, Abruf: 24.12.2011